

Particle Filter-Based On-Line Estimation of Spot Volatility with Nonlinear Market Microstructure Noise Models

Rainer Dahlhaus
(University of Heidelberg)

Jan C. Neddermeyer
(DZ BANK AG and University of Heidelberg)

March 2011

Summary. A new technique for the on-line estimation of spot volatility for high-frequency data is developed. The algorithm works directly on the transaction data and updates the volatility estimate immediately after the occurrence of a new transaction. Furthermore, a new nonlinear market microstructure noise model is proposed that reproduces several stylized facts of high-frequency data. A computationally efficient particle filter is used that allows for the approximation of the unknown efficient prices and, in combination with a recursive EM algorithm, for the estimation of the volatility curve. We neither assume that the transaction times are equidistant nor do we use interpolated prices. We also make a distinction between volatility per time unit and volatility per transaction and provide estimators for both.

Keywords. Nonlinear state-space model; microstructure noise; sequential EM algorithm; sequential Monte Carlo; tick-by-tick data; transaction time.

1 Introduction

In the last couple of years the modeling of financial data observed at high-frequency became one of the major research topics in the field of financial econometrics. It is of high practical relevance because a rising number of market participants execute trades based on high-frequency strategies and are exposed to high-frequency market risk. Examples of those trading strategies are statistical arbitrage, the execution of large block trades, and market making. For most strategies the spot volatility is important for trading signal generation and risk management. Often the immediate detection of sudden volatility movements is particularly relevant for traders. Usually high-frequency trading strategies are highly automated. In fact, they are often “speed games” and only profitable if one reacts to market changes faster than other market participants. In a high-frequency setting the estimation of spot volatility is much more complicated due to the presence of market microstructure noise. Overall, this causes the need for an on-line spot volatility estimator which filters out market microstructure noise and adapts to volatility movements quickly. In addition, it is required to be computationally efficient. We propose an estimation method which satisfies these requirements.

In our method described below, the efficient log-price process of a security is treated as a latent state in a nonlinear state-space model. The relation between the efficient log-prices and the transaction prices is described by a new class of nonlinear market microstructure noise models leading to a particular form of the observation equation in the state-space model. A computationally efficient particle filter is developed which allows the estimation of the filtering distributions of the efficient log-prices given the observed transaction prices. Based on the filtering distributions the time-varying volatility is estimated by using a new sequential Expectation-Maximization (EM) algorithm. Our procedure works on-line and updates the volatility estimate immediately when a new transaction comes in. The method is suitable for real-time applications because of its computational efficiency. Contrary to several other papers we do not assume that the transaction times are equidistant nor do we use interpolated prices.

Until recently, the main focus in the literature has been on the estimation of the integrated volatility. This task has been studied extensively under various assumptions on the market microstructure noise (Zhou 1996; Zhang et al. 2005; Andersen et al. 2006; Bandi and Russell 2006, 2008; Hansen and Lunde 2006; Barndorff-Nielsen et al. 2008; Kalnina and Linton 2008; Christensen et al. 2009; Jacod et al. 2009; Podolskij and Vetter 2009). Some authors suggested that estimates of the spot volatility can be obtained through localized versions of estimators for the integrated volatility (Harris 1990; Foster and Nelson 1996; Zeng 2003; Fan and Wang 2008; Bos et al. 2009; Kristensen 2009) or by (Fourier-) series methods (Munk and Schmidt-Hieber 2009). However, these methods are essentially off-line procedures.

In this article, transaction data are treated as noisy observations of a latent efficient log-price process X_t . We assume that transaction prices Y_{t_j} are observed at times $t_1 < t_2 < \dots < t_T$.

The evolution of the efficient log-price process is modeled by a random walk in transaction time with possibly time-varying volatility σ_{t_j} , that is

$$X_{t_j} = X_{t_{j-1}} + Z_{t_j} \quad (1)$$

with $Z_{t_j} \sim \mathcal{N}(0, \sigma_{t_j}^2)$, or alternatively by a diffusion model in clock time – see Section 3.4. Drift terms are ignored because their effect is of lower order with high-frequency data.

We make a distinction between volatility per time unit and volatility per transaction and provide estimators for both. We start with a model in transaction time instead of clock time leading to an estimator of the spot volatility per transaction. In Section 3.4, a transformation from transaction time volatility to clock time volatility is given leading to a subsequent estimator of the volatility per time unit. In addition, we give a direct clock time estimator. In our opinion a model in transaction time has at least two advantages: First, we are convinced that the returns of the unobserved (log-) efficient price in a transaction time model can be modeled in most situations quite well by a Gaussian distribution (we believe that the often observed “jumps” in asset prices are for high-frequency data often due to an increase of the trading intensity and not to a heavy tailed distribution – see sections 3.4, 5.2, and 6), and second, volatility in transaction time is more constant than volatility in clock time making the algorithm more stable (Ané and Geman 2000; Plerou et al. 2001; Gabaix et al. 2003 – see also Section 5.2). We emphasize that these modeling issues need a deeper investigation. Note that we assume a Gaussian distribution for the unobserved efficient log-price while the observed price is not Gaussian but usually discrete.

The relation between the unobserved efficient (log-)prices and the observed transaction prices is described through a general nonlinear market microstructure noise model which is completely different from the models considered so far. It can be expressed through a nonlinear equation

$$Y_{t_j} = g_{t_j}(\exp[X_{t_j}]) = g_{t_j; Y_{t_1:j-1}}(\exp[X_{t_j}]), \quad (2)$$

where the function g_{t_j} is a (possibly time-inhomogeneous) rounding scheme. A simple example is the rounding of $\exp[X_{t_j}]$ to the nearest cent or the situation where the next trade is made with probability $1/2$ on the closest bid- and with probability $1/2$ on the closest ask-level of an order book. In Section 2 we give several examples. Usually the rounding also depends on past observations $Y_{t_1:j-1} := \{Y_{t_1}, \dots, Y_{t_{j-1}}\}$ or exogenous variables such as order book data or market maker quotes.

The state equation (1) and the observation equation (2) form a nonlinear state-space model (see also (3) and (5)). The spot volatility curve is considered as a parameter of this state-space model. The estimation is done through a particle filter and a new sequential EM-type algorithm. Very roughly speaking our volatility estimator can be viewed as a localized realized volatility estimator based upon the particles of the particle filter. In detail the situation is however more complicated because we need a back and forth between particle filter and volatility estimator to obtain a decent on-line estimator.

The article is organized as follows. Section 2 describes the nonlinear market microstructure noise model. In Section 3, a particle filter and a sequential EM-type algorithm are proposed for on-line estimation of spot volatility in transaction time. Furthermore, the relation between clock time volatility and volatility in transaction time are sketched and two estimates of spot volatility in clock time are given. A description of the implementation of our algorithm is given in Section 4. Finally, simulation results and an application to real data are presented in Section 5 followed by some conclusions in Section 6.

In some parts of the paper we could replace the notation y_{t_j}, x_{t_j} by the simpler notation y_j, x_j etc. Since we need the former notation in Section 3.4 and at a few other places we decided to stick with this notation throughout.

2 Nonlinear Market Microstructure Noise Models

In most existing market microstructure models the efficient log-price is assumed to be corrupted by additive stationary noise (Aït-Sahalia et al. 2005; Zhang et al. 2005; Bandi and Russell 2006; Hansen and Lunde 2006; Barndorff-Nielsen et al. 2008). The noise variables are typically independent of the efficient log-price process. The major weakness of these models is that they cannot reproduce the discreteness of transaction prices. More adequate models which incorporate rounding noise have also been considered (Ball 1988; Large 2007; Li and Mykland 2007; Robert and Rosenbaum 2008; Rosenbaum 2009). Popular models are based on additive noise followed by rounding according to the smallest tick size as in (4). We discuss these models in relation to our model below. Rounding observation noise is also an issue in many other fields such as engineering. An example are roundoff errors due to the word-length restriction which have also been discussed within state-space models (c.f. Hinamoto et al. 2003; Renfors 1983).

Our model is based on the idea that the observed price is obtained from the unknown efficient price by means of a generalized rounding function as in (2). This rounding may be time-inhomogeneous and done in different ways - examples are rounding to the nearest cent, rounding to the closest liquid level of an order book (cf. Example 2 below), rounding to the quotes created by a market maker (cf. Example 3) or to some levels estimated from previous observations $Y_{t_{1:j-1}}$ (cf. Example 4). In general these levels are created from additional exogeneous information or in an unspecified way from previous observations $y_{t_{1:j-1}}$ (e.g. the levels in an order book). In most cases we assume that (conditional on $y_{t_{1:j-1}}$) these levels are known. Example 4 is an example where such exogeneous information is unavailable and the levels are estimated.

Model assumption 1 (observation equation / microstructure noise model):

- (i) The distribution of $Y_{t_j} = g_{t_j; Y_{t_{1:j-1}}}(\exp[X_{t_j}])$ is discrete.
- (ii) The conditional distribution of Y_{t_j} given the state X_{t_j} and previous observations is of the form

$$p(y_{t_j} | y_{t_{1:j-1}}, \exp[x_{t_j}]) \propto \mathbf{1}_{A_{t_j}}(\exp[x_{t_j}]) \quad \text{a.s.} \quad (3)$$

where the set A_{t_j} depends on y_{t_j} and on the conditioning observations $y_{t_1}, \dots, y_{t_{j-1}}$. “a.s.” means almost surely with respect to the distribution of X_{t_j} .

It is important to note that the concrete specification of the set A_{t_j} is an important part of the microstructure noise model at hand. In Examples 1-4 we shall give different options how to specify A_{t_j} in different situations. Note that we need to know A_{t_j} only for the observed y_{t_j} and not for all possible realizations of Y_{t_j} . If the function $g_{t_j} = g_{t_j; Y_{t_1:j-1}}$ is deterministic then $A_{t_j} := g_{t_j}^{-1}(y_{t_j}) = \{z : g_{t_j}(z) = y_{t_j}\}$ is the inverse image of y_{t_j} under g_{t_j} . The “a.s.” will be omitted in the rest of the paper. In particular Proposition 1 continues to hold if (3) only holds almost surely.

Reasons for choosing the model: We chose the above model for three reasons:

- (a) The particle filter takes a simple form: The optimal proposal becomes a truncated normal distribution and the importance weights are also easy to calculate (see Proposition 1). This means that the filter is much more efficient than in the general case and less particles (and less computation time) are needed.
- (b) The model covers many important cases of microstructure noise (see the Examples below). We advocate in particular the deterministic rounding in Example 1 (with its more sophisticated variants in the subsequent examples) which describe in a sufficient way several stylized facts of high-frequency data - namely the discreteness of prices, the bid-ask bounce and time-varying bid-ask spread and the form of autocorrelations and partial autocorrelations of real log-returns (see the simulation example at the end of this chapter).
- (c) Our model is more parsimonious than other models. Popular models with rounding which have recently been considered (not in combination with particle filtering) are

$$Y_{t_j} = \text{round}(\exp[X_{t_j}] + U_{t_j}) \quad \text{or} \quad Y_{t_j} = \text{round}(\exp[X_{t_j} + U_{t_j}]) \quad (4)$$

with a simple rounding function (e.g. to the nearest cent) and for example i.i.d. Gaussian U_{t_j} . It is obvious that these models also do not describe perfectly the underlying mechanism. The drawback of these models is that it is necessary to estimate the variance of the noise variable U_{t_j} while in our model there is no need for estimation of further parameters. We show in the remark at the end of Section 3.1 that these models can also be handled with the methods of this paper if the modified state variable $\tilde{X}_{t_j} = (X_{t_j}, U_{t_j})'$ is used.

Identifiability and approximation of the filtering distribution: (9) and (3) imply that the joint filtering distribution $p(\mathbf{x}_{t_1:j} | \mathbf{y}_{t_1:j})$ is uniquely defined under the model assumptions. A_{t_j} and $\log A_{t_j}$ are the supports of the filtering distribution of the efficient price $p(\exp[x_{t_j}] | \mathbf{y}_{t_1:j})$ and the efficient log-price $p(x_{t_j} | \mathbf{y}_{t_1:j})$, respectively. It will be shown in Section 3.1 that the filtering distributions can be approximated through a particle filter. A real data example is given in Figure 1. It shows the supports A_{t_j} (gray vertical lines) and kernel density estimates of the filtering distributions of the

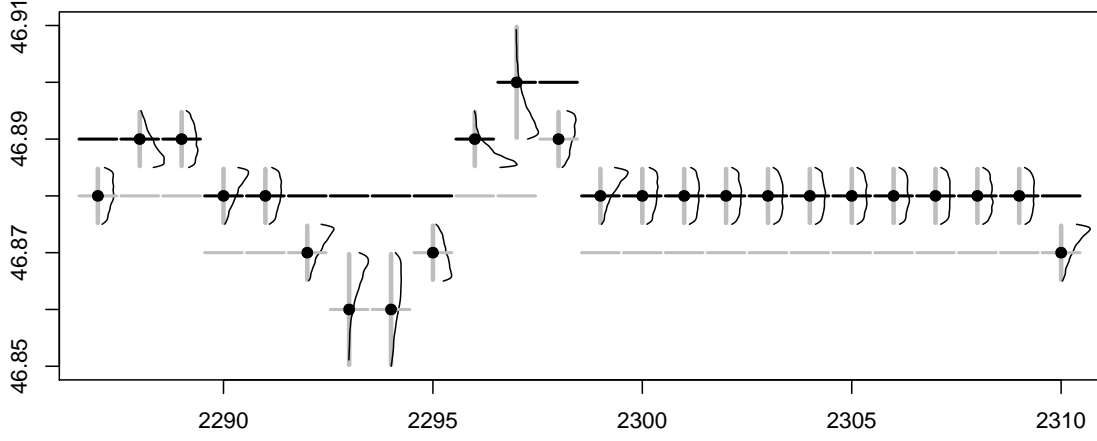


Figure 1: A real data example of estimated filtering distributions based on our market microstructure noise model with deterministic rounding for the case when market maker quotes are available in addition to the transaction data. The details are provided in Section 5.2. The plot shows some transaction prices (circles) along with kernel density estimates of the filtering distributions of the efficient prices (black lines) based on the particles produced by our particle filter. The gray vertical lines indicate the assumed support of the filtering distributions. The bid and ask market maker quotes are displayed by gray and black horizontal lines, respectively. The x-axis shows transaction time.

efficient prices (black lines) which are computed based on the output of the particle filter. In this example, market maker quotes are available (see Example 4 below). The details of this example are provided in Section 5.2.

For completeness we give the state equation again. The reasons for choosing this model for the efficient price have already been given in the introduction.

Model assumption 2 (state equation / efficient price model):

The unobserved efficient price is given by $\exp(x_{t_j})$ with

$$p(x_{t_j} | x_{t_{j-1}}) = \mathcal{N}(x_{t_j} | x_{t_{j-1}}; \sigma_{t_j}^2), \quad (5)$$

$\sigma_{t_j}^2$ is assumed to be either constant or slowly varying in time, that is we assume some smoothness for $\sigma_{t_j}^2$.

The smoothness assumption does not need to be specified any further because we do not use it formally. However, without this assumption the estimation procedure developed in Section 3.2 would not make sense.

Modeling microstructure noise via the specification of A_{t_j} : In order to carry out the particle filter and the volatility estimate described later we have to specify for the observation y_{t_j} at hand the set A_{t_j} i.e. the set of the possible efficient prices. This specification is an important modeling step. We now give examples.

Example 1 (simple deterministic and stochastic rounding):

- (i) The simplest example is the rounding of $\exp[x_{t_j}]$ to the nearest integer (say cent) – i.e. $y_{t_j} = \text{round}(\exp[x_{t_j}])$. In this case $A_{t_j} = [y_{t_j} - 0.5, y_{t_j} + 0.5]$ and $p(y_{t_j} | y_{t_{j-1}}, \exp[x_{t_j}]) = \mathbf{1}_{A_{t_j}}(\exp[x_{t_j}])$.
- (ii) A simple stochastic example is where we choose for $\exp[x_{t_j}] \in (n, n + 1)$ the values $y_{t_j} = n$

and $y_{t_j} = n + 1$ each with probability $1/2$. In that case $A_{t_j} = (y_{t_j} - 1, y_{t_j} + 1)$ and $p(y_{t_j} | y_{t_{1:j-1}}, \exp[x_{t_j}]) = \frac{1}{2} \mathbf{1}_{A_{t_j}}(\exp[x_{t_j}])$ for almost all x_{t_j} . It seems natural to set $y_{t_j} = n$ for $\exp[x_{t_j}] = n$ but with order book data as in Example 2 below this choice is no longer natural.

Surprisingly, it turns out that deterministic rounding describes better the empirically observed behavior of transaction prices than stochastic rounding (see Figure 3 and the discussion in Example 2). For this reason we use throughout this paper different variants of deterministic rounding.

We now give some examples where we model the bid-ask spread of financial transaction data. In these cases A_{t_j} depends on past observations and/or exogenous data.

Example 2 (order book data):

Let's assume that at each transaction time t_j the exchange provides a limit order book with bid and ask levels given by $\alpha_{t_j}^k$ ($k = 1, 2, \dots, K$) and $\beta_{t_j}^\ell$ ($\ell = 1, 2, \dots, L$) respectively (these are the levels where contract offers are really available). The order book levels satisfy $\alpha_{t_j}^K < \dots < \alpha_{t_j}^2 < \alpha_{t_j}^1 < \beta_{t_j}^1 < \beta_{t_j}^2 < \dots < \beta_{t_j}^L$ and we denote

$$\mathcal{M}_{t_j^-} := \{\alpha_{t_j}^K, \dots, \alpha_{t_j}^2, \alpha_{t_j}^1, \beta_{t_j}^1, \beta_{t_j}^2, \dots, \beta_{t_j}^L\}.$$

$\mathcal{M}_{t_j^-}$ represents the state of the order book immediately before the transaction at time t_j occurs. $\mathcal{M}_{t_j^-}$ depends in an unknown way on the past observations $y_{t_{1:j-1}}$ and exogenous information. Clearly $y_{t_j} \in \mathcal{M}_{t_j^-}$. We now set, corresponding to the deterministic case (i) in Example 1

$$A_{t_j} := \{z \in \mathbb{R} : \operatorname{argmin}_{\gamma \in \mathcal{M}_{t_j^-}} |z - \gamma| = y_{t_j}\} \quad (6)$$

or equivalently

$$g_{t_j; y_{t_{1:j-1}}}(z) := \operatorname{argmin}_{\gamma \in \mathcal{M}_{t_j^-}} |z - \gamma|.$$

Thus the transaction price at time t_j is that price from $\mathcal{M}_{t_j^-}$ with the smallest Euclidean distance to the efficient price. This means that the efficient price at time t_j is assumed to be closer to the observed price y_{t_j} than to any other order book level. Of course, this cannot be guaranteed and it seems to be more realistic to choose for (say) $\gamma \in (\alpha_{t_j}^1, \beta_{t_j}^1)$ $y_{t_j} = \alpha_{t_j}^1$ and $y_{t_j} = \beta_{t_j}^1$ each with probability $1/2$ - i.e. a trade is made with probability $1/2$ on the bid and on the ask side. This corresponds to the stochastic case (ii) from Example 1 leading to the definition $A_{t_j} := (\text{largest level from } \mathcal{M}_{t_j^-} \text{ below } y_{t_j}, \text{ smallest level from } \mathcal{M}_{t_j^-} \text{ above } y_{t_j})$. Figure 3 however reveals that the model with A_{t_j} as in (6) (deterministic rounding) much better captures the stylized facts of real transaction data. The explanation may be that in the case of a liquid order book often several trades are executed at the same level and therefore the first model gives a better fit. The situation may be different if the stock is less heavily traded - however we have not investigated that.

In the present situation we could better write instead of (3)

$$p(y_{t_j} | \mathcal{M}_{t_j^-}, \exp[x_{t_j}]) \propto \mathbf{1}_{A_{t_j}}(\exp[x_{t_j}]) \quad \text{a.s.}$$

where $\mathcal{M}_{t_j^-}$ contains implicitly the relevant information from $y_{t_{1:j-1}}$.

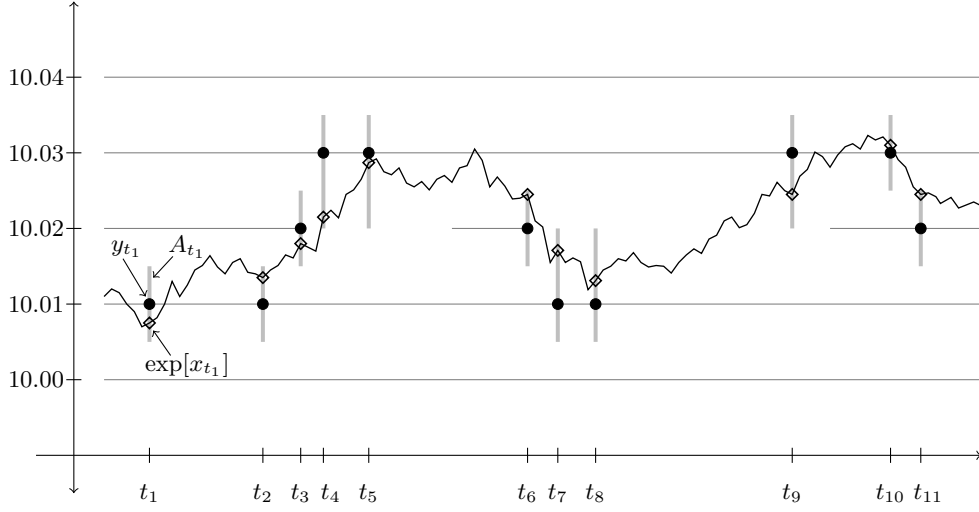


Figure 2: An example of the market microstructure noise model with deterministic rounding for the case when order book data are available. The figure shows the transaction prices (circles), the (in practice unknown) efficient prices in transaction time (diamonds), the latent efficient price process in clock time (black line), the order book levels (gray horizontal lines), and the supports of the filtering distributions of the efficient prices (gray vertical lines).

If the volume of the trade at time t_j is so large that it is executed on several levels of the order book then y_{t_j} should be set equal to the largest ask level (smallest bid level) and all lower levels should be deleted before determining A_{t_j} .

An example of this market microstructure model is visualized in Figure 2. The intervals A_{t_j} are denoted by thick vertical lines. Note, that these are also the supports of the filtering distributions. Larger intervals A_{t_j} are usually due to a larger bid-ask spread.

Example 3 (market maker quotes):

In case where market maker quotes are available instead of order book data, we only have a single bid and a single ask level α_{t_j} and β_{t_j} , respectively, which satisfy $\alpha_{t_j} < \beta_{t_j}$. That is, y_{t_j} is either equal to α_{t_j} or equal to β_{t_j} . Corresponding to the deterministic case in Example 1 we set

$$A_{t_j} = [y_{t_j} - \Delta_{t_j}, y_{t_j} + \Delta_{t_j})$$

where $\Delta_{t_j} := 0.5(\beta_{t_j} - \alpha_{t_j})$.

From a certain point of view this choice of A_{t_j} seems to be not adequate and one is tempted to chose

$$A_{t_j} = \begin{cases} (-\infty, \alpha_{t_j} + \Delta_{t_j}), & \text{if } y_{t_j} = \alpha_{t_j} \\ [\beta_{t_j} - \Delta_{t_j}, \infty), & \text{if } y_{t_j} = \beta_{t_j} \end{cases}.$$

To understand why this is not a proper choice one needs to look in more detail at the behavior of the market maker. Of course the market maker has more (invisible) levels which are automatically executed at the same time if the efficient price makes larger jumps. Furthermore, the market maker has additional information on the efficient price (say from trades of correlated securities) and may have already adjusted his levels towards the efficient price. This last fact violates our model assumptions (in that the function g_{t_j} not only depends on past values $y_{t_{1:j-1}}$ and exogenous information but also somehow on $\exp[x_{t_j}]$) but in particular in this situation our model with the above choice of A_{t_j} seems to be a reasonable parsimonious model.

This example also demonstrates the advantage of the fact that we just have to specify the inverse image A_{t_j} for the y_{t_j} at hand.

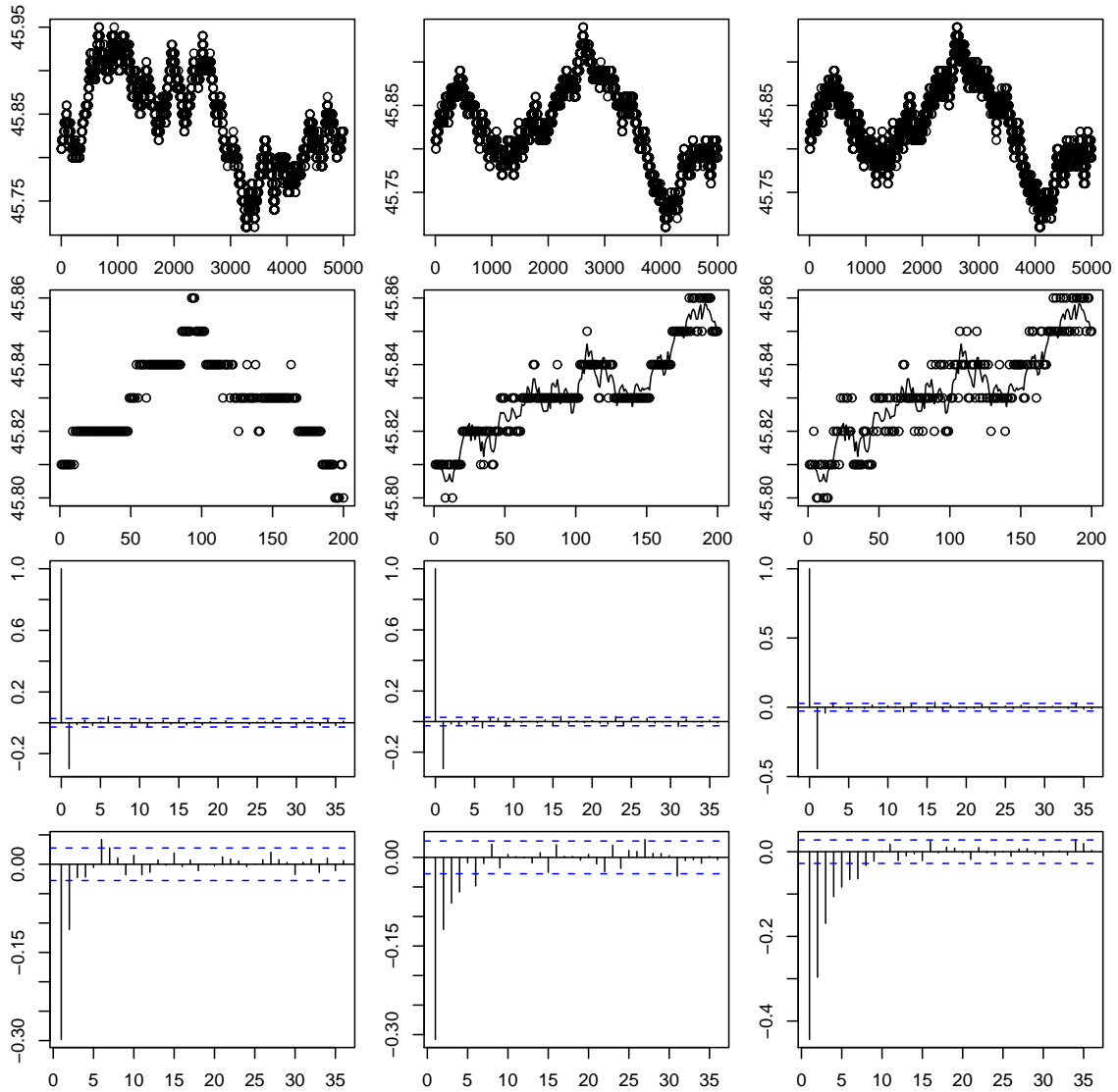


Figure 3: Comparison of real transaction data for Citigroup (left column) with simulated data from the market microstructure noise model with deterministic rounding (middle column) and stochastic rounding (right column). The plots show (from top to bottom): 10,000 transaction prices; the first 250 transaction prices and the efficient price process of the simulated data; the autocorrelations and the partial autocorrelations of the returns of the transaction prices.

Example 4 (transaction data only):

We now consider the situation where no order book data or market maker quotes are available. In this case we try to estimate the order book levels from the data and use afterwards the A_{t_j} from Example 2. More precisely we estimate half the bid-ask spread at time t_j by

$$\Delta_{t_j} = \begin{cases} 0.5 |y_{t_j} - y_{t_{j-1}}| & \text{if } y_{t_j} \neq y_{t_{j-1}}, \\ \Delta_{t_{j-1}} & \text{else} \end{cases}$$

and set

$$A_{t_j} = [y_{t_j} - \Delta_{t_j}, y_{t_j} + \Delta_{t_j}).$$

Surprisingly, this specification does not belong to a deterministic but to a stochastic mapping g_{t_j}

(it is not difficult to see that the same x_{t_j} lies in A_{t_j} constructed from different y_{t_j} - hence g_{t_j} must be stochastic). The mapping g_{t_j} becomes deterministic (conditionally on $y_{t_{1:j-1}}$) if one replaces Δ_{t_j} by $\Delta_{t_{j-1}}$. However $\Delta_{t_{j-1}}$ obviously is much worse than Δ_{t_j} as an estimate of the bid-ask spread at time t_j - so from a practical point of view the above specification is to be preferred.

We finally demonstrate in a simulation example that our model reproduces the autocorrelations and partial autocorrelations of real log-returns. In addition we compare the deterministic and the stochastic rounding from Example 1. In Figure 3 transaction data of Citigroup are compared with data simulated from our model with the two different rounding schemes from Example 1(i) and (ii), respectively. The figure shows the simulated efficient and the observed prices. The efficient log-prices were generated according to (1) such that the observations have approximately the same volatility as the Citigroup data. We emphasize the large number of zero returns in the true data which are reproduced by the simulated data with deterministic rounding. The important point, however, is that our market microstructure noise model with deterministic rounding automatically introduces autocorrelations and partial autocorrelations of the log-returns which are very similar to those of the real Citigroup data.

For this reason, the price discreteness, and the fact that our model covers bid-ask bounces and time-varying bid-ask spreads we believe that our model better describes the real world market microstructure than most existing models.

3 On-Line Estimation of Spot Volatility

We now present on-line algorithms for the estimation of the spot volatility. Because all results also hold in the multivariate case with synchronous trading times we formulate this section for multivariate security prices. We are aware of the fact that the main challenge in the multivariate case are non-synchronous trading times. The presented results are, however, the basis for future work on non-synchronous trading.

We therefore consider in this section the estimation of the covariance matrix Σ_{t_j} which gives the volatilities of the individual efficient log-price processes $\mathbf{X}_t = (X_{t,1}, \dots, X_{t,S})^T$ as well as their cross-volatilities. The algorithms for the spot volatility are obtained by setting $\Sigma_{t_j} = \sigma_{t_j}^2$. The multivariate version of the nonlinear state-space model (3) and (5) is given by

$$p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \exp[\mathbf{x}_{t_j}]) \propto \mathbf{1}_{\mathbf{A}_{t_j}}(\exp[\mathbf{x}_{t_j}]) \quad \text{a.s.}, \quad (7)$$

$$p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}) = \mathcal{N}(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}; \Sigma_{t_j}), \quad (8)$$

where $\mathbf{X}_t = (X_{t,1}, \dots, X_{t,S})^T$ and $\mathbf{Z}_{t_j} \sim \mathcal{N}(\mathbf{0}, \Sigma_{t_j})$. The set \mathbf{A}_{t_j} usually is of the form $\mathbf{A}_{t_j} = A_{t_j,1} \times \dots \times A_{t_j,S}$. We assume that model assumption 1 holds for all components. For simplicity we assume as an initial condition that given $Y_{t_1,s} = y_{t_1,s}$ the efficient prices $\exp[X_{t_1,s}]$ are uniformly distributed on $A_{t_1,s}$.

We remark that (7) and (8) constitute a slightly generalized state-space model because the observations \mathbf{Y}_{t_j} are not conditional independent of $\mathbf{Y}_{t_{1:j-1}}$ given \mathbf{X}_{t_j} as in standard state-space models. However, this is a standard extension which does not cause any difficulty for estimation.

Our objective is the estimation of the covariance matrix Σ_{t_j} based on the observed prices $\mathbf{y}_{t_{1:j}}$. Because of the nonlinear market microstructure noise this is difficult. It is well known that crude estimators that ignore the noise lead to severely biased estimates (see, for instance, Voev and Lunde 2007). The idea of our estimation procedure is to approximate the conditional distribution of the efficient log-prices \mathbf{X}_{t_j} given all observed transaction prices $\mathbf{y}_{t_{1:j}}$ up to time t_j by an efficient particle filter. Based on this approximation a localized EM-type algorithm is used to construct an estimator of Σ_{t_j} .

3.1 An Efficient Particle Filter

Particle filters are sequential Monte Carlo methods (Doucet et al. 2001) that approximate the posterior distributions $p(\mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j}})$ with clouds of particles $\{\mathbf{x}_{t_{1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$. A particle consists of a sample $\mathbf{x}_{t_{1:j}}^i$ and an associated weight $\omega_{t_j}^i$. The particle approximation of the target distribution is given by

$$p(\mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j}}) \approx \sum_{i=1}^N \omega_{t_j}^i \delta_{\mathbf{x}_{t_{1:j}}^i}(\mathbf{x}_{t_{1:j}}),$$

with δ being the Dirac delta function. A particle filter generates particles sequentially in time making use of the relation

$$p(\mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j}}) = \frac{p(\mathbf{y}_{t_j}, \mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j-1}})}{p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}})} = \frac{p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}) p(\mathbf{x}_{t_j}|\mathbf{x}_{t_{j-1}})}{p(\mathbf{y}_{t_j}|\mathbf{x}_{t_{j-1}})} p(\mathbf{x}_{t_{1:j-1}}|\mathbf{y}_{t_{1:j-1}}) \quad (9)$$

and a general sampling technique known as importance sampling. Importance sampling is necessary because direct sampling from (9) is not feasible. In standard state-space models $p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j})$ further simplifies to $p(\mathbf{y}_{t_j}|\mathbf{x}_{t_j})$. As a result of the violated conditional independence property mentioned earlier, this is not the case here.

In each iteration of the particle filter samples are drawn from an importance sampling distribution called proposal. Subsequently, the samples are weighted such that they approximate the target distribution. The choice of the proposal is crucial for the efficiency of the filter. In our framework it is possible to sample from the proposal $p(\mathbf{x}_{t_j}|\mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}})$ which is the optimal proposal in the sense that it minimizes the variance of the importance sampling weights (Doucet et al. 2000). The algorithm can be stated as follows: Assume that weighted particles $\{\mathbf{x}_{t_{1:j-1}}^i, \omega_{t_{j-1}}^i\}_{i=1}^N$ approximating $p(\mathbf{x}_{t_{1:j-1}}|\mathbf{y}_{t_{1:j-1}})$ are given; then

- For $i = 1, \dots, N$:
 - Sample $\mathbf{x}_{t_j}^i \sim p(\mathbf{x}_{t_j}|\mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}^i)$.
 - Compute importance weights

$$\omega_{t_j}^i \propto \omega_{t_{j-1}}^i \frac{p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}^i) p(\mathbf{x}_{t_j}^i|\mathbf{x}_{t_{j-1}}^i)}{p(\mathbf{x}_{t_j}^i|\mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}^i)} = \omega_{t_{j-1}}^i p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_{j-1}}^i).$$

- For $i = 1, \dots, N$:
 - Normalize importance weights $\omega_{t_j}^i = \check{\omega}_{t_j}^i / (\sum_{k=1}^N \check{\omega}_{t_j}^k)$.
- Obtain particles $\{\mathbf{x}_{t_{1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$ which approximate $p(\mathbf{x}_{t_{1:j}} | \mathbf{y}_{t_{1:j}})$.

It is well-known that this algorithm suffers from weight degeneracy which means that after some iterations only few particles will have significant weight. This issue can be resolved by introducing a resampling step that maps the particle system $\{\mathbf{x}_{t_{1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$ onto an equally weighted particle system $\{\mathbf{x}_{t_{1:j}}^i, 1/N\}_{i=1}^N$. Because resampling is time-consuming, it is carried out only if the effective sample size

$$\text{ESS}(\{\omega_{t_j}^i\}_{i=1}^N) = \frac{1}{\sum_{i=1}^N (\omega_{t_j}^i)^2}$$

is below some threshold (Kong et al. 1994). Other resampling schemes are discussed in Douc et al. (2005).

To apply this particle filter to the state-space model given by (7) and (8) it is necessary to specify the optimal proposal and the computation of the importance weights. The following result shows that both take a very simple form.

Proposition 1. *The optimal proposal is a truncated multivariate normal distribution given by*

$$p(\mathbf{x}_{t_j} | \mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}) \propto \mathcal{N}(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}; \Sigma_{t_j})|_{\log \mathbf{A}_{t_j}}$$

with $\log \mathbf{A}_{t_j} = \log A_{t_j,1} \times \dots \times \log A_{t_j,S}$ and the importance weights can be computed through

$$\check{\omega}_{t_j}^i \propto \omega_{t_{j-1}}^i \int_{\log \mathbf{A}_{t_j}} \mathcal{N}(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}^i; \Sigma_{t_j}) d\mathbf{x}_{t_j}. \quad (10)$$

Proof: It is easy to verify that the optimal proposal satisfies

$$p(\mathbf{x}_{t_j} | \mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}) \propto p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}) p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}).$$

leading together with (7) and (8) to the assertion. The expression for the importance weights follows from

$$p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_{j-1}}^i) = \int p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}) p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}^i) d\mathbf{x}_{t_j} = \int_{\log \mathbf{A}_{t_j}} p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}^i) d\mathbf{x}_{t_j}.$$

Remark: For the market microstructure noise models (4) the same methods can be applied after a modification of the state variable. Consider in the univariate case for example $Y_{t_j} = \text{round}(\exp[X_{t_j} + U_{t_j}])$ and let $\mathbf{A}_{t_j} = [y_{t_j} - 0.5, y_{t_j} + 0.5)$.

Consider the corresponding state space models with state variable $\tilde{X}_{t_j} = (X_{t_j}, U_{t_j})'$. Then the optimal proposal satisfies

$$\begin{aligned} p(\tilde{x}_{t_j} | y_{t_{1:j}}, \tilde{x}_{t_{j-1}}) &\propto p(y_{t_j} | y_{t_{1:j-1}}, \tilde{x}_{t_j}) p(\tilde{x}_{t_j} | \tilde{x}_{t_{j-1}}) \\ &= p(y_{t_j} | y_{t_{1:j-1}}, \tilde{x}_{t_j}) p(x_{t_j} | x_{t_{j-1}}) p(u_{t_j}) \\ &= p(x_{t_j} | x_{t_{j-1}}) p(u_{t_j}) \mathbf{1}_{\mathbf{A}_{t_j}}(\exp[x_{t_j} + u_{t_j}]) \end{aligned}$$

which can be used easily to sample the particles in the filter step. The importance weights are given by

$$\begin{aligned}
p(y_{t_j} | y_{t_{1:j-1}}, \tilde{x}_{t_{j-1}}^i) &= p(y_{t_j} | y_{t_{1:j-1}}, x_{t_{j-1}}^i) \\
&= \int p(y_{t_j} | y_{t_{1:j-1}}, \tilde{x}_{t_j}) p(\tilde{x}_{t_j} | x_{t_{j-1}}^i) d\tilde{x}_{t_j} \\
&= \iint \mathbf{1}_{\log \mathbf{A}_{t_j}}(x_{t_j} + u_{t_j}) p(x_{t_j} | x_{t_{j-1}}^i) p(u_{t_j}) dx_{t_j} du_{t_j} \\
&= \iint \mathbf{1}_{\log \mathbf{A}_{t_j}}(x_{t_j} + u_{t_j}) \mathcal{N}(x_{t_j} | x_{t_{j-1}}^i; \sigma_{t_j}^2) \mathcal{N}(u_{t_j} | 0; \sigma_U^2) dx_{t_j} du_{t_j}
\end{aligned}$$

which are however more difficult to compute (compare Section 4).

3.2 A Sequential EM-Type Algorithm

In this section, we discuss the estimation of Σ_{t_j} in the time-constant and time-varying case.

A stochastic EM algorithm can be used to obtain the maximum likelihood estimator in the time-constant case $\Sigma_{t_j} = \Sigma$ (Dempster et al. 1977). The EM algorithm maximizes the likelihood $p_{\Sigma}(\mathbf{y}_{t_{1:T}})$ by iteratively carrying out an E-step and an M-step. In the E-step, the expectation

$$\begin{aligned}
\mathcal{Q}(\Sigma | \hat{\Sigma}^{(m)}) &= \mathbf{E}_{\hat{\Sigma}^{(m)}} [\log p_{\Sigma}(\mathbf{X}_{t_{1:T}}, \mathbf{y}_{t_{1:T}}) | \mathbf{y}_{t_{1:T}}] \\
&= \sum_{j=1}^T \mathbf{E}_{\hat{\Sigma}^{(m)}} [\log p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{X}_{t_j}) | \mathbf{y}_{t_{1:T}}] + \mathbf{E}_{\hat{\Sigma}^{(m)}} [\log p(\mathbf{X}_{t_1}) | \mathbf{y}_{t_{1:T}}] \\
&\quad + \sum_{j=2}^T \mathbf{E}_{\hat{\Sigma}^{(m)}} [\log p_{\Sigma}(\mathbf{X}_{t_j} | \mathbf{X}_{t_{j-1}}) | \mathbf{y}_{t_{1:T}}]
\end{aligned} \tag{11}$$

needs to be approximated, where $\hat{\Sigma}^{(m)}$ is the current estimator. Note, it is sufficient to consider the sum in (11) because the random variables $\log p(\mathbf{y}_{t_j} | \mathbf{y}_{t_{1:j-1}}, \mathbf{X}_{t_j})$ and $p(\mathbf{X}_{t_1})$ do not depend on Σ . In the M-step, a new parameter estimate $\hat{\Sigma}^{(m+1)}$ is obtained by maximizing $\mathcal{Q}(\Sigma | \hat{\Sigma}^{(m)})$.

If Σ_{t_j} is time-varying some regularization is needed. For example $\hat{\Sigma}_{t_j}^{(m+1)}$ can be obtained by maximizing some localized version of (11), e.g.

$$\mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_{1:T}}^{(m)}) = \frac{1}{T} \sum_{k=j-T}^{j-2} \frac{1}{b} K\left(\frac{k}{bT}\right) \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}^{(m)}} [\log p_{\Sigma}(\mathbf{X}_{t_{j-k}} | \mathbf{X}_{t_{j-k-1}}) | \mathbf{y}_{t_{1:T}}] \tag{12}$$

with a kernel $K(\cdot)$ and a bandwidth b .

An approximation of $\mathcal{Q}(\Sigma | \hat{\Sigma}^{(m)})$ and $\mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_{1:T}}^{(m)})$ can be computed based on the smoothing particles

$$\{\mathbf{x}_{t_{1:T}}^i, \omega_{t_T}^i\}_{i=1}^N$$

from our particle filter or (with higher precision) from existing particle smoothing algorithms (Godsill et al. 2004; Neddermeyer 2010; Briers et al. 2010). The smoothing particles give the approximation

$$\begin{aligned}
&\mathbf{E}_{\hat{\Sigma}_{t_{1:T}}^{(m)}} [\log p_{\Sigma}(\mathbf{X}_{t_{j-k}} | \mathbf{X}_{t_{j-k-1}}) | \mathbf{y}_{t_{1:T}}] \\
&\approx \sum_{i=1}^N \omega_{t_T}^i \frac{1}{2} \left[S \log 2\pi + \log |\Sigma| + \text{tr} \left\{ \Sigma^{-1} (\mathbf{x}_{t_{j-k}}^i - \mathbf{x}_{t_{j-k-1}}^i) (\mathbf{x}_{t_{j-k}}^i - \mathbf{x}_{t_{j-k-1}}^i)^T \right\} \right]
\end{aligned} \tag{13}$$

which leads, with

$$\check{\Sigma}_{t_j}(\omega_{t_T}) := \sum_{i=1}^N \omega_{t_T}^i (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i) (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)^T, \quad (14)$$

to the maximizers

$$\hat{\Sigma}^{(m+1)} = \frac{1}{T-1} \sum_{j=2}^T \check{\Sigma}_{t_j}(\omega_{t_T}) \quad (15)$$

and

$$\hat{\Sigma}_{t_j}^{(m+1)} = \left[\sum_k K\left(\frac{k}{bT}\right) \right]^{-1} \sum_k K\left(\frac{k}{bT}\right) \check{\Sigma}_{t_{j-k}}(\omega_{t_T}) \quad (16)$$

of (11) and (12), respectively (note that the particles and, therefore, also $\check{\Sigma}$ depend on m .)

Instead of these estimates, one will prefer in most situations an on-line algorithm which updates the estimates when a new observation comes in. This requires on the one hand the use of filtering particles instead of smoothing particles and on the other hand an integration of the E-step into the algorithm.

We now develop such an algorithm step-by-step. Note that the recursion developed in 1) below is not an on-line algorithm. It is just discussed to demonstrate the relation of the on-line algorithms in (25) and (26) to the estimates (15) and (16), respectively. Note, in the following steps the notation $\hat{\Sigma}_{t_j}$ is used for different estimates.

1) A “recursive” solution for the above situation (both for time-constant and time-varying Σ_{t_j}) is

$$\mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_{1:T}}) := \{1 - \lambda_j\} \mathcal{Q}_{t_{j-1}}(\Sigma | \hat{\Sigma}_{t_{1:T}}) + \lambda_j \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}} [\log p_{\Sigma}(\mathbf{X}_{t_j} | \mathbf{X}_{t_{j-1}}) | \mathbf{y}_{t_{1:T}}] \quad (17)$$

with $\mathcal{Q}_{t_2}(\Sigma | \hat{\Sigma}_{t_{1:T}}) = \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}} [\log p_{\Sigma}(\mathbf{X}_{t_2} | \mathbf{X}_{t_1}) | \mathbf{y}_{t_{1:T}}]$ leading to

$$\begin{aligned} \mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_{1:T}}) &= \sum_{k=0}^{j-3} \left[\prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}} [\log p_{\Sigma}(\mathbf{X}_{t_{j-k}} | \mathbf{X}_{t_{j-k-1}}) | \mathbf{y}_{t_{1:T}}] \\ &\quad + \left[\prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}} [\log p_{\Sigma}(\mathbf{X}_{t_2} | \mathbf{X}_{t_1}) | \mathbf{y}_{t_{1:T}}]. \end{aligned} \quad (18)$$

With the “constant parameter setting” $\lambda_j := 1/(j-1)$, where $\left[\prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} = \frac{1}{j-1}$, this gives the classical (quasi-) likelihood

$$\frac{1}{j-1} \sum_{k=0}^{j-2} \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}} [\log p_{\Sigma}(\mathbf{X}_{t_{j-k}} | \mathbf{X}_{t_{j-k-1}}) | \mathbf{y}_{t_{1:T}}],$$

that is (11) for $j = T$. Furthermore, the maximizer of (18) is, with the smoother-approximation as in (13) and $\check{\Sigma}_{t_j}(\omega_{t_T})$ as in (14), given by

$$\hat{\Sigma}_{t_j}^{(m+1)} = \sum_{k=0}^{j-3} \left[\prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} \check{\Sigma}_{t_{j-k}}(\omega_{t_T}) + \left[\prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] \check{\Sigma}_{t_2}(\omega_{t_T}). \quad (19)$$

This can be written as the recursion

$$\hat{\Sigma}_{t_j}^{(m+1)} = \{1 - \lambda_j\} \hat{\Sigma}_{t_{j-1}}^{(m+1)} + \lambda_j \check{\Sigma}_{t_j}(\omega_{t_T})$$

with $\hat{\Sigma}_{t_2}^{(m+1)} = \check{\Sigma}_{t_2}(\omega_{t_T})$. Again, we obtain with the “constant parameter setting” $\lambda_j := 1/(j-1)$ that $\hat{\Sigma}_{t_j}^{(m+1)}$ coincides with the estimate in (15) for $j = T$.

2) On-line algorithms: The above algorithm is not an on-line algorithm because the conditional expectation in (17) depends on all observations. Therefore, we replace the conditioning set of variables $\{\mathbf{y}_{t_{1:T}}\}$ by $\{\mathbf{y}_{t_{1:j}}\}$ meaning that we pass from the smoothing distribution to the filtering distribution. More precisely, $\mathbf{E}_{\hat{\Sigma}_{t_{1:T}}}[\log p_{\Sigma}(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_{1:T}}]$ is replaced by $\mathbf{E}_{\hat{\Sigma}_{t_{1:j-1}}}[\log p_{\Sigma}(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_{1:j}}]$ (we need at this point an estimate for Σ_{t_j} - see the comment at the end of this section) leading to the on-line algorithm

$$\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:j-1}}) := \{1 - \lambda_j\} \mathcal{Q}_{t_{j-1}}(\Sigma|\hat{\Sigma}_{t_{1:j-2}}) + \lambda_j \mathbf{E}_{\hat{\Sigma}_{t_{1:j-1}}}[\log p_{\Sigma}(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_{1:j}}] \quad (20)$$

with $\mathcal{Q}_{t_2}(\Sigma|\hat{\Sigma}_{t_1}) = \mathbf{E}_{\hat{\Sigma}_{t_1}}[\log p_{\Sigma}(\mathbf{X}_{t_2}|\mathbf{X}_{t_1})|\mathbf{y}_{t_{1:2}}]$. (18) holds analogously and we now obtain analogous to (19) the estimate

$$\hat{\Sigma}_{t_j} = \sum_{k=0}^{j-3} \left[\prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} \check{\Sigma}_{t_{j-k}}(\omega_{t_{j-k}}) + \left[\prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \right] \check{\Sigma}_{t_2}(\omega_{t_2}) \quad (21)$$

now with

$$\check{\Sigma}_{t_j}(\omega_{t_j}) := \sum_{i=1}^N \omega_{t_j}^i (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)^T \quad (22)$$

based on the filtering particles $\{\mathbf{x}_{t_{j-1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$. This estimate can be obtained from the on-line recursion

$$\hat{\Sigma}_{t_j} = \{1 - \lambda_j\} \hat{\Sigma}_{t_{j-1}} + \lambda_j \check{\Sigma}_{t_j}(\omega_{t_j}) \quad \text{with} \quad \hat{\Sigma}_{t_2} = \check{\Sigma}_{t_2}(\omega_{t_2}). \quad (23)$$

Observe that the estimated covariance matrix $\hat{\Sigma}_{t_j}$ is positive semi-definite by construction.

The new parameter estimate $\hat{\Sigma}_{t_j}$ is used afterwards to calculate the next filtering particles and their weights $\{\mathbf{x}_{t_{j+1}}^i, \omega_{t_{j+1}}^i\}_{i=1}^N$ followed by the calculation of $\hat{\Sigma}_{t_{j+1}}$ via another application of (23) etc. In contrast to the standard EM algorithm, our sequential variant therefore updates the covariance estimate (which in turn is used in the next step of the particle filter) in every time step. In the “new E-step”, $\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:j-1}})$ is approximated through

$$\begin{aligned} \hat{\mathcal{Q}}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:j-1}}) &= \{1 - \lambda_j\} \hat{\mathcal{Q}}_{t_{j-1}}(\Sigma|\hat{\Sigma}_{t_{1:j-2}}) \\ &\quad - \lambda_j \frac{1}{2} \sum_{i=1}^N \omega_{t_j}^i \left[S \log 2\pi + \log |\Sigma| + \text{tr} \left\{ \Sigma^{-1} (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)^T \right\} \right] \end{aligned} \quad (24)$$

using the particles $\{\mathbf{x}_{t_{j-1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$ which are generated as described in Section 3.2. In the “new M-step”, the maximization of $\hat{\mathcal{Q}}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:j-1}})$ gives the on-line estimator defined in (23).

Note that $\check{\Sigma}_{t_j}(\omega_{t_j})$ is not an approximation of the conditional variance $\text{Var}(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}}|\mathbf{y}_{t_{1:j}})$ but an approximation of $\mathbf{E}((\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})^2|\mathbf{y}_{t_{1:j}})$ (both are different because $\mathbf{E}(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}}|\mathbf{y}_{t_{1:j}}) \neq 0$). As a result of $\mathbf{E}[\mathbf{E}((\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})^2|\mathbf{Y}_{t_{1:j}})] = \mathbf{E}(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})^2 = \text{Var}(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})$, $\hat{\Sigma}_{t_j}$ is a descent estimator of Σ_{t_j} .

3) Time-constant covariance matrices: If Σ_{t_j} is time-constant the first idea is to apply the algorithm (23) with the “constant parameter setting” $\lambda_j = 1/(j-1)$. However, the situation is different from the classical case in that the “old” estimate $\hat{\Sigma}_{t_{j-1}}$ has in addition some bias due to the use of particles generated with an estimated covariance instead of the true one. Therefore we need to put less weight on the first term in (23). The situation has been carefully investigated for a similar algorithm in the i.i.d.-case by Cappé and Moulines (2009). Following their recommendation we use in our situation the on-line algorithm

$$\hat{\Sigma}_{t_j} = \{1 - (j-1)^{-\gamma}\} \hat{\Sigma}_{t_{j-1}} + (j-1)^{-\gamma} \check{\Sigma}_{t_j}(\omega_{t_j}) \quad (25)$$

with $\gamma \in (\frac{1}{2}, 1)$. Cappé and Moulines prove consistency and asymptotic normality of their estimate for weights $\lambda_j := \lambda_0 j^{-\gamma}$ and $\gamma \in (\frac{1}{2}, 1)$ and also for $\gamma = 1$ under some restrictions on λ_0 (Theorem 2). Furthermore, in their simulations it turned out that a value of $\gamma = 0.6$ and $\lambda_0 = 1$ has lead to good estimates. From our experience we prefer the choice $\gamma = 0.9$ and $\lambda_0 = 1$ (see Figure 5). Even-Dar and Mansour (2003) obtained an optimal value of about 0.85 in a related estimation problem. We emphasize that the choice of γ needs more investigations - both theoretical and practical.

4) Time-varying covariance matrices: If Σ_{t_j} is time-varying it is necessary to put more weight on recent observations. In this case, the traditional solution is to use the algorithms (17), (20) and (23) with time-constant $\lambda_j \equiv \lambda$ instead of a decaying λ_j . That is we use in the time-varying case

$$\hat{\Sigma}_{t_j} = \{1 - \lambda\} \hat{\Sigma}_{t_{j-1}} + \lambda \check{\Sigma}_{t_j}(\omega_{t_j}) \quad \text{with} \quad \hat{\Sigma}_{t_2} = \check{\Sigma}_{t_2}(\omega_{t_2}) \quad (26)$$

where the choice of λ depends on the smoothness of the true volatility curve (cf. the relation to a kernel estimate below). To adapt locally to this smoothness one may either choose a time varying λ_j anyhow (in some way dependent on the data) or use the SAGES procedure (see Section 4 below) where the algorithm is run simultaneously for K different values of λ and the optimal estimate is determined at each step as a convex combination of these estimates.

For a deeper understanding we stress the following heuristics: If $\lambda_j \equiv \lambda$ and $t_j = j\delta$ (e.g. $\delta = \frac{1}{T}$) then we have with $b := \frac{\delta}{\lambda}$ for $\delta \rightarrow 0$

$$\left[\prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \right] \lambda_{j-k} = (1 - \lambda)^k \lambda = \frac{\delta}{b} \left(1 - \frac{\delta}{b}\right)^{\frac{1}{\delta} k \delta} \approx \frac{\delta}{b} \left(e^{-\frac{1}{b}}\right)^{k \delta} = \frac{\delta}{b} K\left(\frac{k \delta}{b}\right) \quad (27)$$

where $K(x) := e^{-x}$. That is $\mathcal{Q}_{t_j}(\Sigma | \hat{\Sigma}_{t_{1:T}})$ from (20) is basically the kernel likelihood given in (16) with the one-sided exponential kernel, and $\hat{\Sigma}_{t_j}$ given by (26) is basically the kernel estimate

$$\hat{\Sigma}_{t_j} = \left[\sum_k K\left(\frac{k}{bT}\right) \right]^{-1} \sum_k K\left(\frac{k}{bT}\right) \sum_{i=1}^N \omega_{t_j-k}^i (\mathbf{x}_{t_j-k}^i - \mathbf{x}_{t_j-k-1}^i) (\mathbf{x}_{t_j-k}^i - \mathbf{x}_{t_j-k-1}^i)^T.$$

3.3 Combining the Particle Filter and the Sequential EM-Type Algorithm

To summarize our estimation method consists of three components:

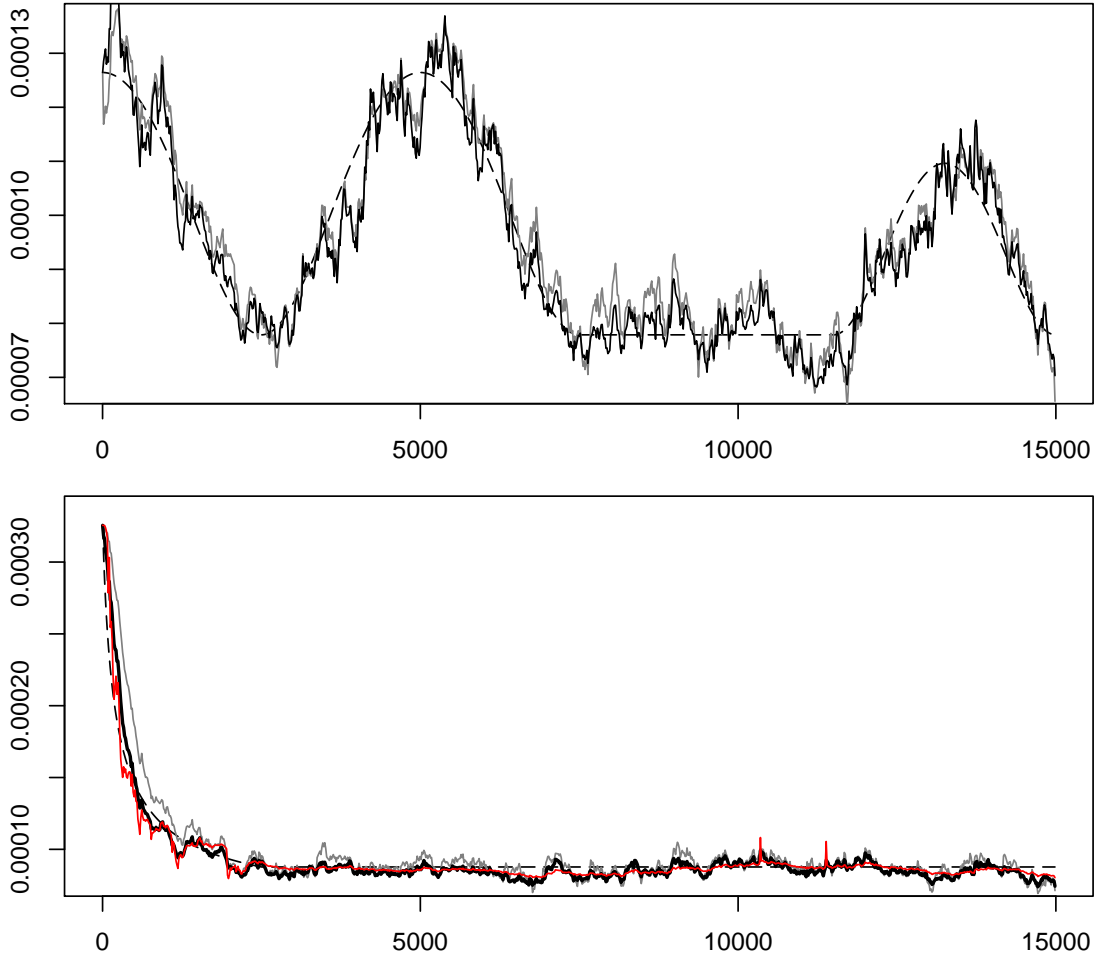


Figure 4: Estimation of two time-varying volatility curves given by the dashed lines based on simulated data. Estimators: our estimator $\hat{\Sigma}_{t_j}$ (black line), our estimator combined with SAGES for adaptive step size selection $\hat{\Sigma}_{t_j}^S$ (red line), and a benchmark estimator (gray line). For details see Section 5.1.

- (i) The state-space model with a new market microstructure noise model and the transaction time model for the efficient log-price ((7) and (8));
- (ii) A particle filter which sequentially approximates the filtering distributions of the efficient log-prices given the observed transaction prices (Section 3.1);
- (iii) The on-line EM-type estimator $\hat{\Sigma}_{t_j}$ given by (25) or (26) which estimates Σ_{t_j} based on the particle approximation of the filtering distribution obtained from the particle filter (Section 3.2).

A key aspect of the method is the back and forth between the particle filter and the EM-type estimator. To propagate the particles from time t_j to time t_{j+1} the particle filter requires an estimator of $\Sigma_{t_{j+1}}$ which we denote by $\hat{\Sigma}_{t_{j+1}}^{\text{pf}}$. A simple solution is to use $\hat{\Sigma}_{t_{j+1}}^{\text{pf}} := \hat{\Sigma}_{t_j}$ from the previous EM-type step. The EM-type estimator then in turn updates the covariance estimate based on the new particles for time t_{j+1} generated by the particle filter.

Estimation results of our estimator $\hat{\Sigma}_{t_j}$ and a benchmark estimator (see Section 5.1) are presented in Figure 4. Details and a discussion are given in Section 5.1.

3.4 From Transaction Time to Clock Time

Clock Time Spot Volatility Estimation

In the preceding section, we have derived an algorithm for the estimation of the covariance matrix $\Sigma_{t_j} = \Sigma(t_j)$ in a transaction time model. If one prefers a clock time model all results of this paper continue to hold with some modifications. In this case one may consider as the underlying model the stochastic differential equation

$$d\mathbf{X}(t) = \Gamma(t) d\mathbf{W}(t) \quad \text{where} \quad \Gamma(t) \Gamma^T(t) = \Sigma^c(t) \quad (28)$$

and $\mathbf{W}(t)$ is a multivariate Brownian motion. $\Sigma^c(t)$ is the volatility curve in the clock time model. Loosely speaking, it denotes volatility per time unit while $\Sigma(t)$ denotes the volatility per transaction at time t . The relation between the two curves should be given by (30) (of course this depends on the mathematical definition of $\Sigma(t)$ and $\Sigma^c(t)$). If we set $\mathbf{X}_{t_j} = \mathbf{X}(t_j)$ we obtain almost the same state space model as in (7) and (8) but with a modified variance of the transition distribution which is now approximately given by

$$p(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}) = \mathcal{N}(\mathbf{x}_{t_j} | \mathbf{x}_{t_{j-1}}; |t_j - t_{j-1}| \Sigma^c(t_j)).$$

This is the only change needed in the state-space model (7), (8). As an estimate $\hat{\Sigma}_{t_j}^c$ we can use the on-line estimates (25) and (26) but now with the update matrix $\check{\Sigma}_{t_j}(\omega_{t_j})$ replaced by

$$\check{\Sigma}_{t_j}^c(\omega_{t_j}^c) := \sum_{i=1}^N \omega_{t_j}^{ci} \frac{(\mathbf{x}_{t_j}^{ci} - \mathbf{x}_{t_{j-1}}^{ci})(\mathbf{x}_{t_j}^{ci} - \mathbf{x}_{t_{j-1}}^{ci})^T}{|t_j - t_{j-1}|} \quad (29)$$

based on the modified filtering particles $\{\mathbf{x}_{t_{j-1}:j}^{ci}, \omega_{t_j}^{ci}\}_{i=1}^N$.

An Alternative Estimator for Clock Time Spot Volatility

In the diffusion model (28) the spot volatility in clock time is

$$\Sigma^c(t) = \lim_{\Delta t \rightarrow 0} \frac{\int_t^{t+\Delta t} \Sigma^c(s) ds}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{\text{Var}(\mathbf{X}(t + \Delta t) - \mathbf{X}(t))}{\Delta t}.$$

To clarify the relation to the transaction time volatility $\Sigma(t)$ we assume that the transaction times t_j are realizations of a stochastic point process with intensity function $\lambda_I(t)$ (transaction rate) which is independent of the efficient and observed prices. We then have

$$\begin{aligned} \lim_{\Delta t \rightarrow 0} \frac{\text{Var}(\mathbf{X}(t + \Delta t) - \mathbf{X}(t))}{\Delta t} &= \lim_{\Delta t \rightarrow 0} \mathbf{E} \frac{\sum_{j: t < t_j \leq t + \Delta t} \Sigma(t_j)}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \mathbf{E} \frac{\sum_{j: t < t_j \leq t + \Delta t} \Sigma(t_j)}{|\{j : t < t_j \leq t + \Delta t\}|} \frac{|\{j : t < t_j \leq t + \Delta t\}|}{\Delta t} = \Sigma(t) \lambda_I(t) \end{aligned}$$

that is

$$\Sigma^c(t) = \Sigma(t) \lambda_I(t). \quad (30)$$

We stress that this is primarily a nonparametric relation (“variance per time unit = variance per transaction \times expected number of transactions per time unit”) and it depends on the underlying model whether this coincides with the definition of $\Sigma^c(t)$ and $\Sigma(t)$ given in the model. A model which exactly leads to this formula is the subordinated differential equation $d\mathbf{X}(t) = \Gamma(t) d\mathbf{W}_{N(t)}$ with a point process $N(t)$ with intensity $\lambda_I(t)$ (cf. Howison and Lamper 2001). The unit of Δt (e.g. milliseconds) determines the unit of $\Sigma(t)$ (e.g. variance per millisecond) and of the intensity (e.g. expected number of transactions per millisecond). An obvious estimate of the clock time volatility therefore is $\hat{\Sigma}^c(t_j) = \hat{\Sigma}_{t_j} \times |\{\ell : t_j - \Delta t < t_\ell \leq t_j\}| / \Delta t$ with some Δt .

Here we advocate a different estimation method of the intensity function $\lambda_I(t)$ which is closer related to our on-line scheme, namely the estimation of $\lambda_I(t)$ by the inverse of the averaged duration times $\bar{\delta}_j$ defined by the recursion

$$\bar{\delta}_j = (1 - \lambda_j) \bar{\delta}_{j-1} + \lambda_j (t_j - t_{j-1}) \quad \text{with} \quad \bar{\delta}_2 = t_2 - t_1$$

leading with (21) to the alternative clock time volatility estimator

$$\hat{\Sigma}_{\text{alt}}^c(t_j) := \frac{\hat{\Sigma}_{t_j}}{\bar{\delta}_j} = \frac{\lambda \sum_{k=0}^{j-3} (1 - \lambda)^k \check{\Sigma}_{t_{j-k}}(\omega_{t_{j-k}}) + (1 - \lambda)^{j-2} \check{\Sigma}_{t_2}(\omega_{t_2})}{\lambda \sum_{k=0}^{j-3} (1 - \lambda)^k (t_{j-k} - t_{j-k-1}) + (1 - \lambda)^{j-2} (t_2 - t_1)}.$$

This estimator has a remarkable property: Because $\check{\Sigma}_{t_\ell}(\omega_{t_\ell}) \approx (t_\ell - t_{\ell-1}) \check{\Sigma}_{t_\ell}^c(\omega_{t_\ell}^c)$ the estimator is of the form

$$\hat{\Sigma}_{\text{alt}}^c(t_j) \approx \frac{\sum_{k=0}^{j-2} w_k \check{\Sigma}_{t_{j-k}}^c(\omega_{t_{j-k}}^c)}{\sum_{k=0}^{j-2} w_k}$$

that is $\hat{\Sigma}_{\text{alt}}^c(t_j)$ is a weighted average of the $\check{\Sigma}_{t_\ell}^c(\omega_{t_\ell}^c)$ and therefore also a decent estimator in the clock time model (the “ \approx ” signs stem from the fact that in $\check{\Sigma}_{t_\ell}(\omega_{t_\ell})$ and $\check{\Sigma}_{t_\ell}^c(\omega_{t_\ell}^c)$ two different particle filters are used - the effect of this is not clear!). Notice that the denominator $t_{j-k} - t_{j-k-1}$ in $\check{\Sigma}_{t_{j-k}}^c(\omega_{t_{j-k}}^c)$ cancels out - leading therefore to a more stable estimator (for example the sharp black peaks in Figures 7 and 8 are caused by small values of $t_{j-k} - t_{j-k-1}$).

The above argument contains a pitfall: While $\Sigma(t)$ usually is smooth thus requiring a small value of λ , the intensity of the point process $\lambda_I(t)$ changes considerably over time thus requiring larger value of λ . For that reason we use different step sizes λ for the estimators $\hat{\Sigma}_{t_j}$ and $\bar{\delta}_j$ (compare Section 5.2).

4 Implementation Overview

The Algorithm

The particle filter uses the following steps for $j = 2, \dots, T$ (see Proposition 1)

- For $i = 1, \dots, N$:
 - Generate $\mathbf{x}_{t_j}^i$ from the optimal proposal $\mathcal{N}(\mathbf{x}_{t_j}^i | \mathbf{x}_{t_{j-1}}^i; \hat{\Sigma}_{t_j}^{\text{pf}}) |_{\log \mathbf{A}_{t_j}}$ with $\hat{\Sigma}_{t_j}^{\text{pf}} = \hat{\Sigma}_{t_{j-1}}$.
 - Compute the importance weight $\check{\omega}_{t_j}^i$ as in (10). If $S = 1$ this is given by

$$\check{\omega}_{t_j}^i \propto \omega_{t_{j-1}}^i \left\{ \Phi(\sup \log A_{t_j} | \mathbf{x}_{t_{j-1}}^i; \hat{\Sigma}_{t_j}^{\text{pf}}) - \Phi(\inf \log A_{t_j} | \mathbf{x}_{t_{j-1}}^i; \hat{\Sigma}_{t_j}^{\text{pf}}) \right\}.$$

- For $i = 1, \dots, N$: Normalize the importance weight $\omega_{t_j}^i = \check{\omega}_{t_j}^i / \sum_{k=1}^N \check{\omega}_{t_j}^k$.
- If the effective sample size $\text{ESS}(\{\omega_{t_j}^i\}_{i=1}^N) < cN$ (with say $c = 0.2$), then resample the particles using, for instance, the residual resampling scheme (Douc et al. 2005).
- Update the estimator $\hat{\Sigma}_{t_{j-1}}$ according to (25) or (26).

Overall the algorithm is very easy to implement in a few lines. It is computationally very efficient because the complexity of one iteration is linear in the number of particles N . In addition resampling is required only rarely because the optimal proposal is used. In our applications resampling was carried out only about every 15th iteration using a threshold for the effective sample size of $c = 0.2$. As a result of the efficiency of our particle filter, the number of particles N is not a critical quantity. Typically, about 500 particles suffice to achieve a sufficient precision (see Figure 5).

Note that in the multivariate case the sampling from the optimal proposal and the evaluation of the importance weights is nontrivial. However, both the sampling from and the evaluation of a truncated normal distribution are standard problems in statistics which have been discussed extensively in the literature. Relevant references for the sampling problem are Geweke (1991) and Robert (1995). More recent approaches based on Gibbs sampling are described by Kotecha and Djuric (1999) and Rodriguez-Yam et al. (2004). Also for the numerical approximation of multivariate (rectangular) normal probabilities several efficient methods have been proposed for instance by Genz (1992, 2004) and Joe (1995).

Step Size Selection

In the time-constant case we use the decreasing step size $\lambda_j = (j - 1)^{-0.9}$ as proposed in Section 3.2. This choice is empirically justified (see Figure 5).

The step size in the time-varying case is data dependent and can be obtained through the following procedure. The mean squared error of $\hat{\Sigma}_{t_j}$ is minimized with respect to λ by the cross-validation type criterion

$$\text{crit}(\lambda) := \sum_{j=2}^{T-1} (\hat{\Sigma}_{t_j} - \check{\Sigma}_{t_{j+1}}(\omega_{t_{j+1}}))^2. \quad (31)$$

This cannot be done on-line. In practice, one will use in an on-line setting a λ from past experience with similar data sets. The expectation of the above criterion is approximately

$$\sum_{j=2}^{T-1} \left[(\mathbf{E}\hat{\Sigma}_{t_j} - \Sigma_{t_j})^2 + \text{Var}(\hat{\Sigma}_{t_j}) + \text{Var}(\check{\Sigma}_{t_{j+1}}(\omega_{t_{j+1}})) \right].$$

Because the last term does not depend on λ we correctly minimize the approximate mean squared error.

Adaptive Step Size Selection Using SAGES

To adaptively select non-constant step sizes λ_j in the time-varying case we propose to use spatially aggregated exponential smoothing (SAGES) developed by Chen and Spokoiny (2009).

In our setting the SAGES method works as follows. The basic idea is to run K volatility estimators $\hat{\Sigma}_{t_j}^k$ in parallel with different step sizes $\lambda^1 > \lambda^2 > \dots > \lambda^K$. The resulting SAGES estimate $\hat{\Sigma}_{t_j}^S$ is then a convex combination of these estimators. In practice we have, say, $K = 15$ which implies that the computational offset is minimal. In fact, only the recursion (26) needs to be computed K times with different step sizes.

For every time step j the SAGES estimate $\hat{\Sigma}_{t_j}^S$ is obtained from the estimators $\hat{\Sigma}_{t_j}^k$, $k = 1, \dots, K$, through the following recursion.

(i) Set $\hat{\Sigma}_{t_j}^{S,1} = \hat{\Sigma}_{t_j}^1$

(ii) For $k = 2, \dots, K$: Compute

$$\hat{\Sigma}_{t_j}^{S,k} = \left(\frac{\gamma_k}{\hat{\Sigma}_{t_j}^k} + \frac{1 - \gamma_k}{\hat{\Sigma}_{t_j}^{S,k-1}} \right)^{-1},$$

where

$$\gamma_k = K \left(\frac{1}{\kappa_{k-1} \lambda^k} \mathcal{K}(\hat{\Sigma}_{t_j}^k, \hat{\Sigma}_{t_j}^{S,k-1}) \right)$$

with kernels $K(u) = \{1 - (u - 1/6)^+\}^+$ and $\mathcal{K}(\Sigma, \tilde{\Sigma}) = -0.5\{\log(\Sigma/\tilde{\Sigma}) + 1 - \Sigma/\tilde{\Sigma}\}$.

(iii) Obtain the SAGES estimate $\hat{\Sigma}_{t_j}^S = \hat{\Sigma}_{t_j}^{S,K}$.

Note that this method can be applied completely on-line. The parameters $\kappa_1, \kappa_2, \dots, \kappa_{K-1}$ are critical values (independent of the time step j) which can be calculated beforehand through a Monte Carlo simulation. Note that SAGES is a univariate method. For a more detailed description and a theoretical analysis of the SAGES method see Chen and Spokoiny (2009).

Initialization

Our experience from many data sets is that the algorithm stabilizes quickly provided that reasonable starting values are used – e.g. $\hat{\Sigma}_{t_2}$ is chosen from prior knowledge or set to a rough initial estimate. The particle filter is started by simulating the $x_{t_1,s}^i$ such that the $\exp[x_{t_1,s}^i]$ are uniformly distributed on $A_{t_1,s}$. In order to exclude the effect of starting values we have used in the simulations (except from Figure 5) the true matrix Σ_{t_2} as the starting value (i.e. $\hat{\Sigma}_{t_2}^{\text{pf}} = \hat{\Sigma}_{t_2} = \Sigma_{t_2}$).

5 Simulations and Applications

5.1 Results for Simulated Data

Estimation of time-constant spot volatility

We first consider the estimation of time-constant spot volatility. An efficient log-price process is simulated from t_1 to t_{5000} with squared volatility equal to $\Sigma_t = 0.0001^2$. The initial efficient price $\exp[X_{t_1}]$ is sampled from a uniform distribution on $[50 - 0.005, 50 + 0.005]$. The transaction prices are obtained by rounding the efficient prices to the nearest cent (see Example 1 (i) in Section 2).

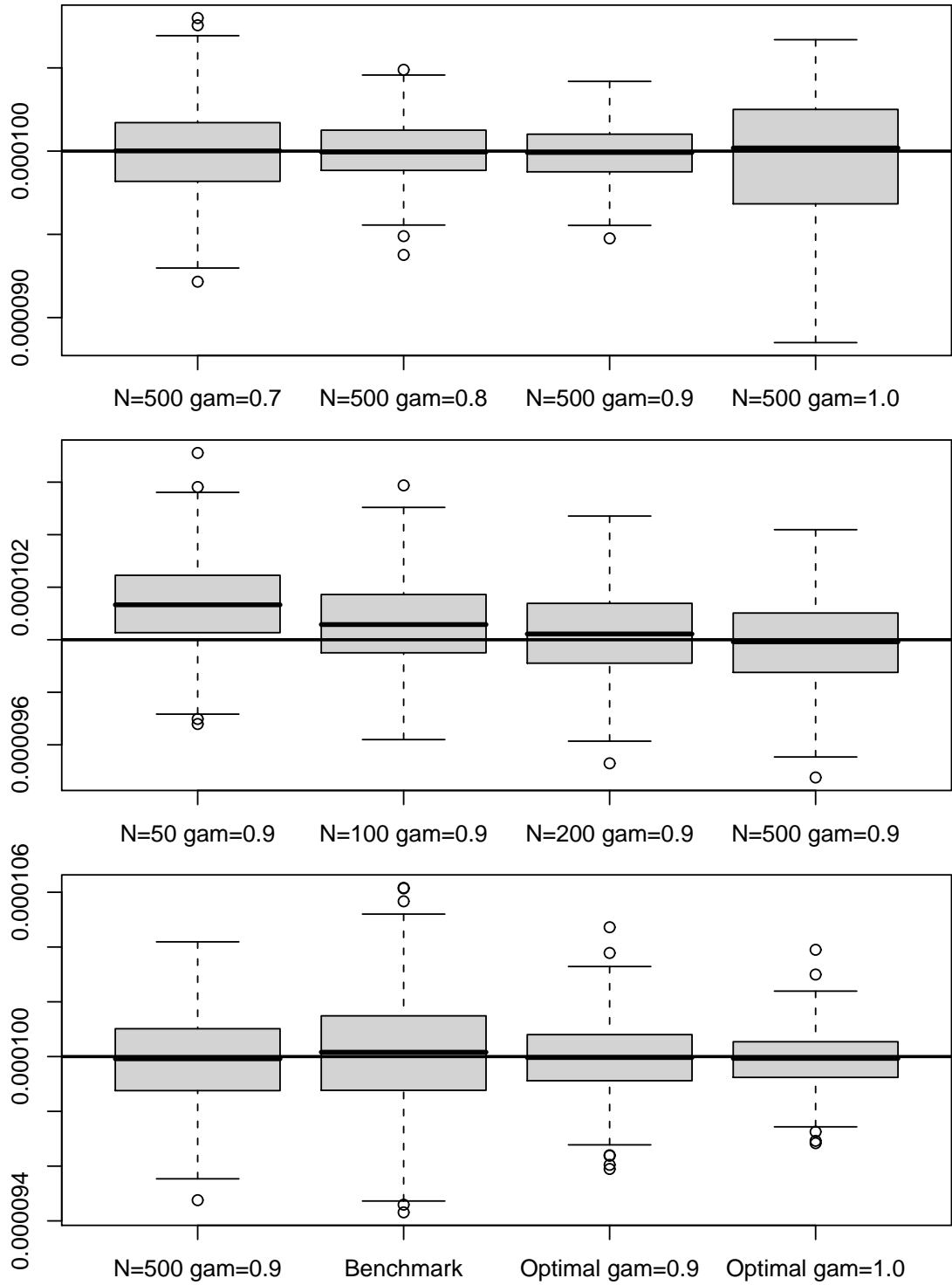


Figure 5: Box plots for the estimation of a time-constant volatility based on simulated data (5,000 transactions). The estimator (25) is applied with different numbers of particles N and different γ and compared to the benchmark estimator and the optimal estimator (not available in practice). The box plots are based on 500 independent runs.

Our algorithm for time-constant spot volatility estimation (25) is applied with different numbers of particles N and different values of γ . The initial value $\hat{\Sigma}_{t_2}^{\text{pf}} = \hat{\Sigma}_{t_2}$ is drawn from a uniform distribution on $(0.00006^2, 0.00014^2)$ which is quite uninformative. For comparison the results of two benchmark algorithms are also reported. The first benchmark method (“Benchmark” in Figure 5) is a recursive estimator with a simpler microstructure noise correction. It is related to the method in Zumbach et al. (2002) and it is based on the market microstructure model $\log Y_{t_j} = X_{t_j} + U_{t_j}$, where the noise variables U_{t_j} are i.i.d. with $\text{Var } U_{t_j} = \eta^2$. The recursive estimator is given by

$$\hat{\Sigma}_{t_j}^{\text{B}} := \left\{1 - \frac{1}{j-1}\right\} (\hat{\Sigma}_{t_{j-1}}^{\text{B}} + \max\{0, 2\hat{\eta}_{t_{j-1}}^2\}) + \frac{1}{j-1} (\log y_{t_j} - \log y_{t_{j-1}})^2 - \max\{0, 2\hat{\eta}_{t_j}^2\} \quad (32)$$

where $\hat{\eta}_{t_j}^2 := \left\{1 - \frac{1}{j-2}\right\} \hat{\eta}_{t_{j-1}}^2 - \frac{1}{j-2} (\log y_{t_j} - \log y_{t_{j-1}}) (\log y_{t_{j-1}} - \log y_{t_{j-2}})$ (here $\frac{1}{j-2}$ is used instead of $\frac{1}{j-1}$ because the algorithm starts one time point later). The term $\max\{0, 2\hat{\eta}_{t_j}^2\}$ corrects for the market microstructure noise. This follows from the fact that

$$\text{Cov}(\log Y_{t_j} - \log Y_{t_{j-1}}, \log Y_{t_{j-1}} - \log Y_{t_{j-2}}) = -\eta^2.$$

The second benchmark method is, in some sense, the optimal estimator (“Optimal” in Figure 5). It is unavailable in practice because it uses the latent efficient log-prices. It is computed analogous to (25) but instead of the particles it employs the efficient log-prices leading to

$$\hat{\Sigma}_{t_j}^{\text{Opt}} = \{1 - (j-1)^{-\gamma}\} \hat{\Sigma}_{t_{j-1}}^{\text{Opt}} + (j-1)^{-\gamma} (x_{t_j} - x_{t_{j-1}})^2.$$

The simulation results are given in terms of box plots which are obtained by 500 independent runs (Figure 5). The box plots suggest that our volatility estimator is asymptotically unbiased and that $\gamma = 0.9$ is a reasonable value. We can also conclude that about 500 particles are sufficient which makes our algorithm computationally efficient and suitable for real-time applications. In addition, it can be observed that the benchmark estimator has a larger variance than our estimator.

Estimation of time-varying spot volatility

We now compare our estimator for time-varying spot volatility $\hat{\Sigma}_{t_j}$ defined in (26) with a benchmark estimator. The efficient log-prices are generated with respect to the time-varying volatility given by the gray dashed lines in Figure 4. The first case (upper plot) is more challenging while the second case (lower plot) is more realistic for a volatility curve in transaction time - see the real data example in Figure 6. In both cases we use for the initial price $\exp[X_{t_1}] \sim \mathcal{U}[50 - 0.005, 50 + 0.005]$. Again transaction prices (observations) are obtained by rounding the efficient prices to the nearest cent. 15,000 transactions are generated which is typical for one trading day of a liquid stock. The particle filter is applied with $N = 500$ particles. Our estimator $\hat{\Sigma}_{t_j}$ uses the constant step size λ obtained by minimizing (31). Analogous to (32) we consider the benchmark estimator given by

$$\hat{\Sigma}_{t_j}^{\text{B}} := \{1 - \lambda\} (\hat{\Sigma}_{t_{j-1}}^{\text{B}} + \max\{0, 2\hat{\eta}_{t_{j-1}}^2\}) + \lambda (\log y_{t_j} - \log y_{t_{j-1}})^2 - \max\{0, 2\hat{\eta}_{t_j}^2\} \quad (33)$$

with $\hat{\eta}_{t_j}^2 := \{1 - \frac{1}{j-2}\} \hat{\eta}_{t_{j-1}}^2 - \frac{1}{j-2} (\log y_{t_j} - \log y_{t_{j-1}})(\log y_{t_{j-1}} - \log y_{t_{j-2}})$. λ is obtained by minimizing the criterion

$$\sum_{j=2}^{T-1} (\hat{\Sigma}_{t_j}^B + \max\{0, 2\hat{\eta}_{t_j}^2\} - (\log y_{t_{j+2}} - \log y_{t_{j+1}})^2)^2 \quad (34)$$

(the terms $\hat{\Sigma}_{t_j}^B + \max\{0, 2\hat{\eta}_{t_j}^2\}$ and $(\log y_{t_{j+2}} - \log y_{t_{j+1}})^2$ are independent in the additive microstructure noise model $\log Y_{t_j} = X_{t_j} + U_{t_j}$ with U_{t_j} i.i.d. - thus by using $(\log y_{t_{j+2}} - \log y_{t_{j+1}})^2$ (34) becomes a decent estimate of the mean squared error (plus a term constant in λ)). For $\hat{\eta}_{t_j}^2$ we use the step sizes $\frac{1}{j-2}$ because η_t^2 should be close to a constant function.

All estimators use the true volatility as starting value. Typical outcomes of the estimators are given in Figure 4. Note that volatility is plotted (instead of squared volatility). In the second case (lower plot) a constant step size is clearly suboptimal. Therefore we also computed our estimator combined with the SAGES method for adaptive step size selection $\hat{\Sigma}_{t_j}^S$ as described in Section 4. $\hat{\Sigma}_{t_j}^S$ is calculated using $K = 15$ step sizes ranging from 0.05 to 0.00005 (equally spaced). In the first case (upper plot) the estimator $\hat{\Sigma}_{t_j}^S$ didn't give better results than the estimator $\hat{\Sigma}_{t_j}$ and is therefore omitted. We also tried to use the SAGES method for the benchmark estimator. This gave surprisingly bad results which are not reported here.

Because the true $\Sigma(t_j)$ is known we can compute the mean squared error $\sum_{j=2}^{T-1} (\hat{\Sigma}(t_j) - \Sigma(t_j))^2$ for the estimators which gives 1.21×10^{-18} and 1.34×10^{-18} for $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}_{t_j}^B$, respectively, for the upper plot in Figure 4. For the estimators $\hat{\Sigma}_{t_j}$, $\hat{\Sigma}_{t_j}^S$, and $\hat{\Sigma}_{t_j}^B$ in the lower plot we obtain 8.59×10^{-19} , 7.52×10^{-19} , and 2.55×10^{-18} . In both plots, our estimators significantly outperforms the benchmark estimator.

The general impression from Figure 4 is that the estimates are undersmoothed. This is in part due to the on-line procedure which corresponds to a one-sided kernel. Additional variability comes in from the particle filter where the estimated covariance matrix is used instead of the true one. We feel that our estimates come close to the need of practitioners who like to have a quickly reacting estimate and who prefer to correct undersmoothed estimates by "eye inspection".

5.2 Results for Real Data

We use stock data from the TAQ data base. Transactions and market maker quotes of the symbol C (Citigroup) for the 3rd September 2007 were extracted from the TAQ data base. To improve the data quality we carried out the following data cleaning and transformation.

Cleaning A: Delete all transactions (quotes) with time stamps outside the main trading period (9:30 AM to 4 PM).

Cleaning B: Delete all transactions (quotes) that are not originating from the NYSE.

Cleaning C: Delete all transactions with abnormal sale condition or corrected prices (see the TAQ User's Guide for details).

Data transformation: If multiple transactions have the same time stamp (after the data cleaning) apply the following transformation. Assume $t_j = t_{j+1} = \dots = t_{k-1} \neq t_k$. Replace t_l by $t'_l = t_j + (l - j)(t_k - t_j)/(k - j)$ for $l = j + 1, \dots, k - 1$.

After the data cleaning 16,287 transactions remained. The transformation replaces identical time stamps with time stamps that are equally spaced. This transformation is necessary because the time stamp precision of our data is limited to one second. We mention that the transformation is only required for clock time volatility estimation. If the volatility is estimated in transaction time then the time stamps are irrelevant (only the order of the transactions matters).

Unfortunately, the quality of the TAQ data is too poor to match easily the transactions with the market maker quotes. Note that it is necessary for our method that the transaction and quote data are perfectly matched. Therefore, our simulations are mainly focused on transaction data.

Estimation results for real market maker quotes

In order to show how our method works in the case when market maker quotes are available (Example 3 in Section 2) we matched by hand (through an adjustment of the time stamps) the quotes and transactions of symbol C for a fraction of the trading day. As mentioned earlier, the quality of our data is too poor to do this automatically. Our particle filter is used with $N = 5,000$ particles to estimate the filtering distributions of the unknown efficient (log-)prices. Figure 1 gives kernel density estimates of filtering distributions of some efficient prices which are computed based on the particle approximations. The market maker quotes, the transaction prices, and supports of the filtering distributions are also shown. From the figure it can be seen that some filtering distributions are highly skewed. In addition, consecutive zero returns lead to very uninformative filtering distributions (see transactions 2,300 through 2,309).

Estimation results for real transaction data

We apply our estimators $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}_{t_j}^S$ with $N = 500$ particles and the benchmark method $\hat{\Sigma}_{t_j}^B$ (33) to estimate the spot volatility for C. An initial volatility of 0.0005 is used.

The transaction data of C and the volatility estimators are shown in Figure 6. At the beginning of the trading day the volatility is large and highly varying. Later, the volatility settles down and seems to be almost constant. Therefore, the SAGES method for localized step size selection is clearly advantageous compared to fixed step sizes. Again the benchmark estimator is rougher than our estimators. Practically, the volatility in transaction time is almost constant after 10:00 AM which in our experience is a typical feature of transaction data of liquid stocks.

Clock time spot volatility estimation

We now compare our two approaches for the estimation of spot volatility in clock time proposed in Section 3.4. The first estimator $\hat{\Sigma}_{t_j}^{cS}$ we consider is the estimator $\hat{\Sigma}_{t_j}^c$ combined with the SAGES

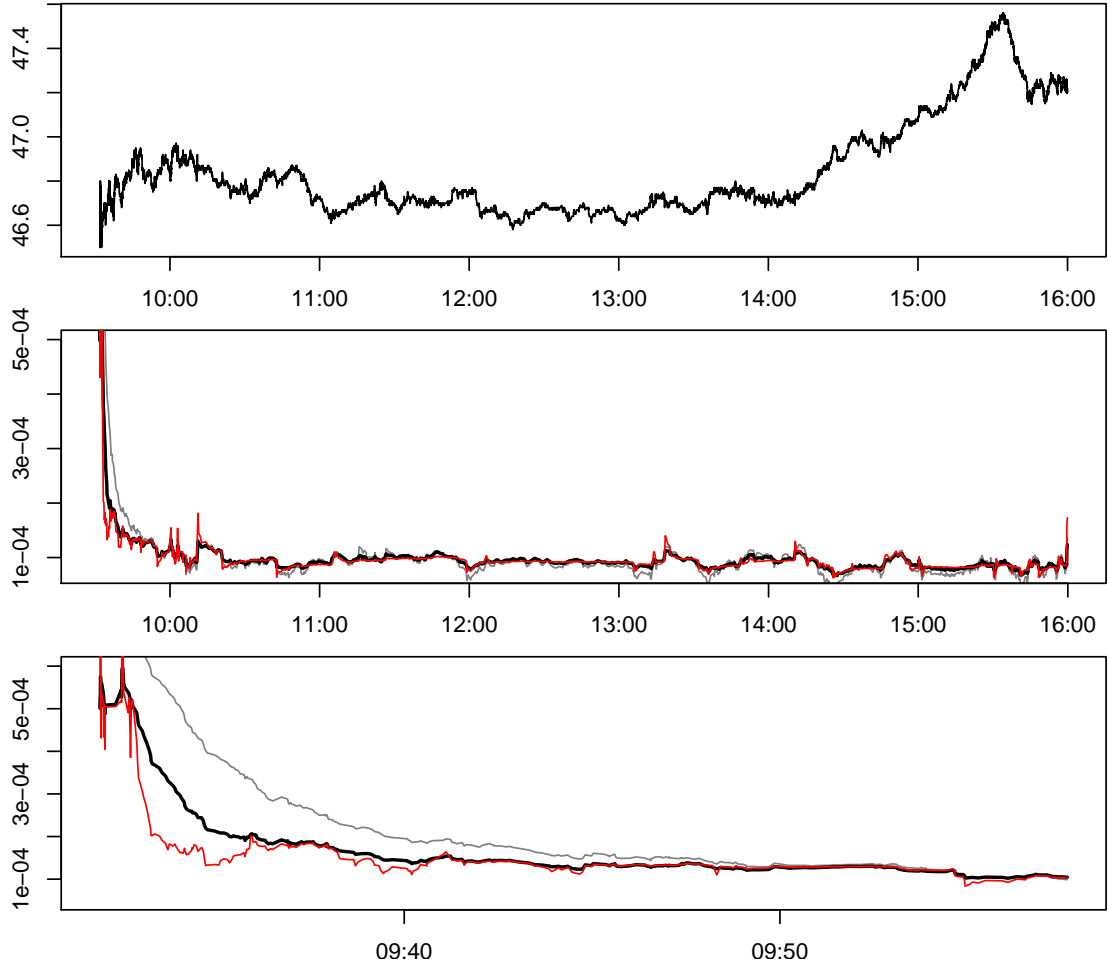


Figure 6: Real data example: Estimation of time-varying spot volatility in transaction time. The upper plot shows the transaction data of the symbol C for the 3rd September 2007. The middle and the lower plot give the volatility estimators $\hat{\Sigma}_{t_j}^S$ (black line), $\hat{\Sigma}_{t_j}^{cS}$ (red line), and the benchmark estimator $\hat{\Sigma}_{t_j}^B$ (gray line).

method (black line in Figure 7). Because the spot volatility in clock time is more volatile than in transaction time SAGES requires larger step sizes. We use step sizes equally spaced between 0.3 and 0.003. The second estimator is the alternative estimator $\hat{\Sigma}_{alt}^{cS}(t_j) = \hat{\Sigma}_{t_j}^S / \bar{\delta}_j$ with the transaction time estimator $\hat{\Sigma}_{t_j}^S$ from Figure 6 (red line). For the duration estimator $\bar{\delta}_j$ we determined the stepsize by minimizing the prediction error $\sum_{j=2}^{T-1} \{\bar{\delta}_j - (t_{j+1} - t_j)\}^2$ leading to $\lambda = 0.1025$. (Because of the dependence of the durations $\bar{\delta}_j$ and $(t_{j+1} - t_j)$ usually are not independent and the minimization of the above criterion therefore is not approximately the same as the minimization of the mean squared error. Despite of this we think that the resulting λ is reasonable.)

The estimation results are provided in Figures 7 and 8. First we state that both estimators roughly coincide in magnitude (which was not clear beforehand). From the upper plot of Figure 7 we observe that $\hat{\Sigma}_{t_j}^{cS}$ (black line) produces some large spikes during the trading day (due to small values of $t_j - t_{j-1}$). The variability of $\hat{\Sigma}_{alt}^{cS}(t_j) = \hat{\Sigma}_{t_j}^S / \bar{\delta}_j$ is mainly a result of the variability of the duration estimator $\bar{\delta}_j$ (plotted in the lower plot) because the transaction time estimator $\hat{\Sigma}_{t_j}^S$ is almost constant (apart from the beginning of the trading day - see Figure 6). The fluctuation of the duration estimator is very high during the whole day.

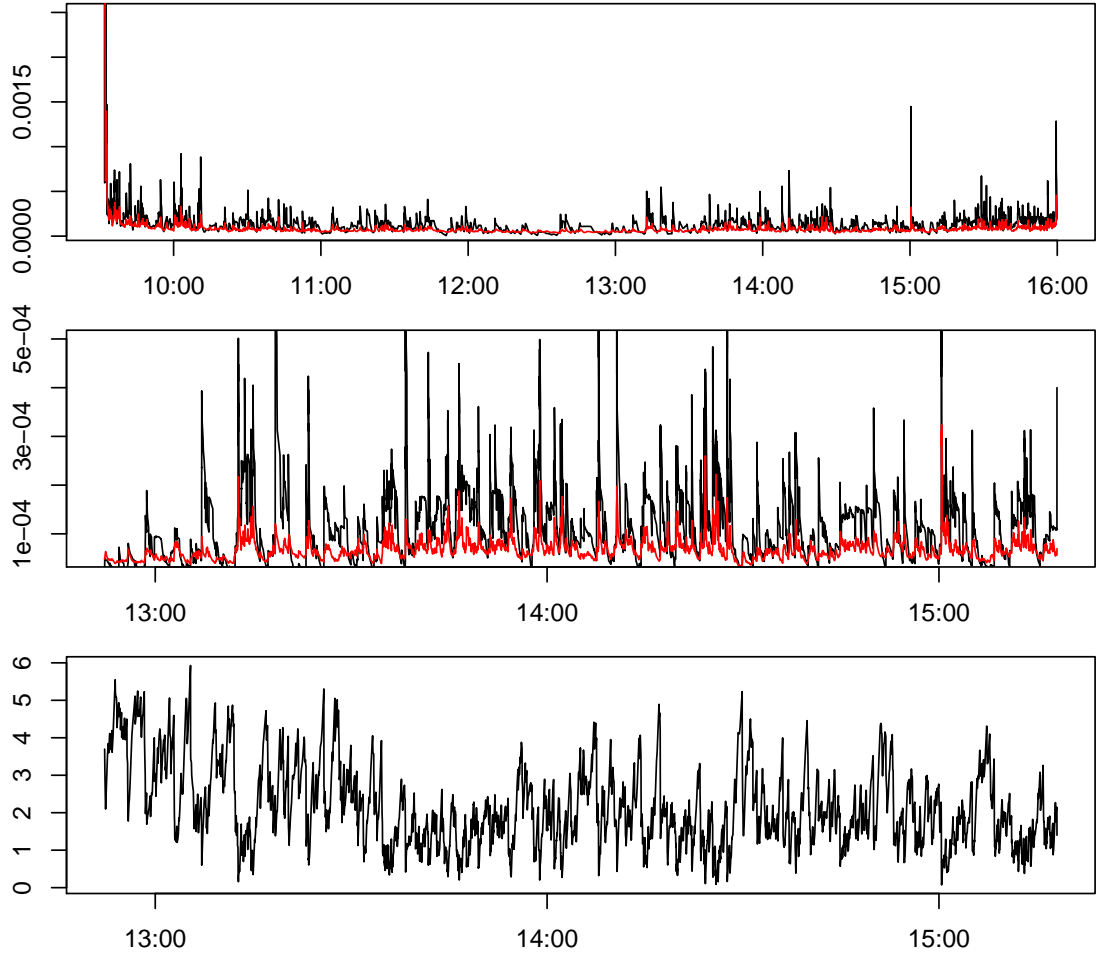


Figure 7: Real data example: Estimation of time-varying spot volatility in clock time based on the transactions of symbol C for the 3rd September 2007. The upper plot gives the volatility estimators $\hat{\Sigma}_{t_j}^{cS}$ (black line) and $\hat{\Sigma}_{alt}^{cS}(t_j)$ (red line). The middle plot shows the estimators for a fraction of the trading day. The averaged duration times $\bar{\delta}_j$ (for a fraction of the trading day) are given in the lower plot (the y-axis shows seconds).

Figure 8 compares the transaction data and the volatility estimates for a small time period. The different behavior of the two estimators is apparent. We regard the strong spikes of $\hat{\Sigma}_{t_j}^{cS}$ as artificial due to small values of $t_j - t_{j-1}$. Note that the estimator needs about one minute to settle down again after the occurrence of a spike. On the other hand the small spikes of $\hat{\Sigma}_{alt}^{cS}(t_j)$ are caused by small averaged durations. For this reason we have more confidence in the second estimator. In addition, it is theoretically more appealing (because the transaction time volatility is almost constant and the variability of the clock time volatility is mainly caused by the variability of the trading intensity).

The second estimator is also more stable for another reason: Because volatility in transaction time is less varying the particle filter in transaction time is more stable.

6 Concluding Remarks

We have presented a new technique for the on-line estimation of time-varying volatility based on noisy transaction data. The algorithm updates the volatility estimate immediately after the

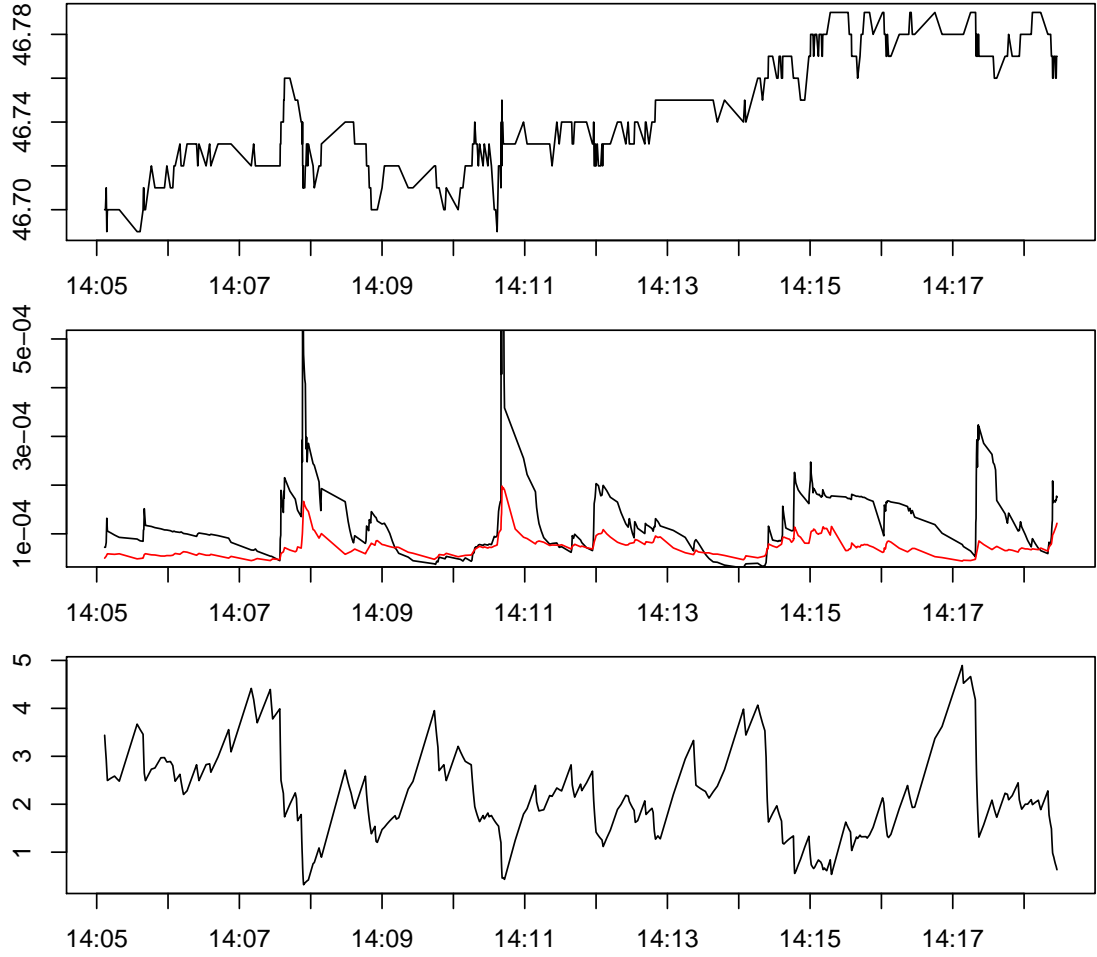


Figure 8: Real data example: Estimation of time-varying spot volatility in clock time based on the transactions of symbol C for the 3rd September 2007. The figure only gives the results for a small fraction of the trading day (compare Figure 7). The plots show (from top to bottom): transaction prices of C; our volatility estimators $\hat{\Sigma}_{t_j}^{cS}$ (black line) and $\hat{\Sigma}_{alt}^{cS}(t_j)$ (red line); the averaged duration times $\bar{\delta}_j$.

occurrence of a new transaction, and it therefore is as close to the market as possible. It is straightforward to extend our method to more complicated price models (e.g. with a drift term) or other microstructure noise models.

Our work was guided by the goal to execute all calculations on-line in a high-frequency situation, and, at the same time, to base all methods on solid statistical principles. We feel that this goal has been achieved: Our algorithm is computationally efficient and it can be applied in real-time. On a recent personal computer an efficient implementation of our method requires a few milliseconds for a single update of the estimator (including one iteration of the particle filter with 500 particles). We use particle filters in nonlinear state space models and EM-type algorithms.

The contribution of this work is manifold. First, we have proposed a nonlinear market microstructure noise model that covers bid-ask bounces, time-varying bid-ask spreads, and the discreteness of prices observed in real data. Second, the problem of on-line volatility estimation has been treated in a nonlinear state-space framework. The filtering distribution of the efficient price can be approximated with a particle filter and the volatility can be estimated as a parameter of the filtering distribution. Third, we have presented a new sequential EM-type algorithm

which allows the on-line estimation of time-varying volatility. The usefulness of the approach for real-time applications has been demonstrated through Monte Carlo simulations and applications to stock data.

Besides the new microstructure noise model we make a clear distinction between the (spot) volatility per time unit $\Sigma^c(t)$ and the volatility per transaction $\Sigma(t)$. Volatility in clock time usually is much more volatile than volatility in transaction time. We advocate the use of transaction time for modeling, i.e. to estimate $\Sigma(t)$, together with a subsequent transformation based on the trading intensity to obtain an estimator for $\Sigma^c(t)$. At least for our data sets it turned out that volatility in transaction time is almost constant (apart from the beginning of the trading day) and the fluctuation of clock time volatility is merely a result of fluctuation of the trading intensity. Thus a new focus in volatility estimation may be on the modeling of trading times.

The empirical finance literature suggests that the bias induced by market microstructure effects makes the most finely sampled data unusable, and most authors prefer to use log-returns with longer lags. Using log-returns with longer lags merely reduces the impact of microstructure, rather than quantifying and correcting its effect. It is obvious that a very good description and removal of microstructure effects allows for using log-returns with lag 1 leading to the most efficient estimate. Our estimates in (22) or (29) use log-returns with lag 1 but they can easily be modified to lag k by setting (in the transaction and continuous time model respectively)

$$\check{\Sigma}_{t_j}(\omega_{t_j}) := \frac{1}{k} \sum_{i=1}^N \omega_{t_j}^i (\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-k}}^i)(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-k}}^i)^T \quad \check{\Sigma}_{t_j}^c(\omega_{t_j}^c) := \sum_{i=1}^N \omega_{t_j}^{ci} \frac{(\mathbf{x}_{t_j}^{ci} - \mathbf{x}_{t_{j-k}}^{ci})(\mathbf{x}_{t_j}^{ci} - \mathbf{x}_{t_{j-k}}^{ci})^T}{|t_j - t_{j-k}|}$$

in case the microstructure correction with the state space model turns out to be insufficient for certain securities (with the same recursions (25) and (26)).

Of course it is desirable to have a complete mathematical theory on the methods of this paper. However, we think that this is very hard to achieve. Mathematically exact are the results on the particle filter given that the true volatility is known (i.e. with $\hat{\Sigma}_{t_j}^{\text{pf}} = \Sigma_{t_j}$) - in particular the results from Proposition 1 on the optimal proposal and the importance weights. This means that the particle filter determines correctly the conditional distribution of the efficient prices given the observations. Even in the case of constant volatility and for the simplest estimator $\hat{\Sigma}_{t_j}$ from (25) it seems to be very difficult to establish consistency and the asymptotic distribution of the volatility estimator. In the simpler context of i.i.d.-observations convergence properties of recursive EM-type algorithms have been studied in Titterton (1984), Sato (2000), Wang and Zhao (2006), and Cappé and Moulines (2009) where also proofs of consistency and asymptotic normality are provided. Furthermore, there are several open mathematical questions on the interplay between transaction time volatility and clock time volatility.

Acknowledgement: We are very grateful to the associate editor and two anonymous referees whose comments helped to improve the paper considerably.

Disclaimer: The views expressed here are those of the authors and not necessarily those of its employers.

References

- Aït-Sahalia, Y., Mykland, P.A., and Zhang, L. (2005) How Often to Sample a Continuous-Time Process in the Presence of Market Microstructure Noise. *Review of Financial Studies*, 18, 351-416.
- Andersen, T.G., Bollerslev, T., and Meddahi, N. (2006) Realized Volatility Forecasting and Market Microstructure Noise. unpublished manuscript.
- Ané, T., and Geman, H. (2000) Order Flow, Transaction Clock, and Normality of Asset Returns. *The Journal of Finance*, 55, 2259-2284.
- Ball, C.A. (1988) Estimation Bias Induced by Discrete Security Prices. *The Journal of Finance*, 43, 841-865.
- Bandi, F.M., and Russell, J.R. (2006) Separating microstructure noise from volatility. *Journal of Financial Economics*, 79, 655-692.
- (2008) Microstructure noise, realized variance, and optimal sampling. *Review of Economic Studies*, 75, 339-369.
- Barndorff-Nielsen, O.E., Hansen, P.R., Lunde, A., and Shephard, N. (2008) Designing Realized Kernels to Measure the Ex-Post Variation of Equity Prices in the Presence of Noise. *Econometrica*, 76, 1481-1536.
- Bos, C.S., Janus, P., and Koopman, S.J. (2009) Spot Variance Path Estimation and its Application to High Frequency Jump Testing. Discussion Paper TI 2009-110/4, Tinbergen Institute.
- Briers, M., Doucet, A., and Maskell, S. (2010) Smoothing Algorithms for State-Space Models. *Annals Institute Statistical Mathematics*, 62, 61-89.
- Cappé, O., and Moulines, E. (2009) On-line expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society, Series B*, 71, 593-613.
- Chen, Y. and Spokoiny, V. (2009) Modeling and estimation for nonstationary time series with applications to robust risk management. unpublished manuscript.
- Christensen, K., Podolskij, M., and Vetter, M. (2009) Bias-correcting the realised range-based variance in the presence of market microstructure noise. *Finance and Stochastics*, 13, 239-268.
- Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39, 1-38.
- Douc, R., Cappé, O., and Moulines, E. (2005) Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, 64-69.
- Doucet, A., Godsill, S., and Andrieu, C. (2000) On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10, 197-208.
- Doucet, A., de Freitas, N., and Gordon, N. (ed.) (2001) *Sequential Monte Carlo Methods in Practice*. New York: Springer.
- Engle, R.F., and Russell, J.R. (1998) Autoregressive conditional duration: A new model for irregularly spaced transaction data. *Econometrica*, 66, 1127-1162.
- Even-Dar, E., and Mansour, Y. (2003) Learning Rates for Q-learning. *Journal of Machine Learning Research*, 5, 1-25.
- Fan, J., and Wang, Y. (2008) Spot volatility estimation for high-frequency data. *Statistics and Its Interface*, 1, 279-288.
- Foster, D., and Nelson, D. (1996) Continuous Record Asymptotics for Rolling Sample Estimators. *Econometrica*, 64, 139-174.

- Gabaix, X., Gopikrishnan, P., Plerou, V., and Stanley, H.E. (2003) A theory of power-law distributions in financial market fluctuations. *Nature*, 423, 267-270.
- Genz, A. (1992) Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1, 141-149.
- Genz, A. (2004) Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing*, 14, 151-160.
- Geweke, J. (1991) Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, Ed. E. Keramidas and S. Kaufman, 571-578. American Statistical Association, Alexandria, VA.
- Godsill, S.J., Doucet, A., and West, M. (2004) Monte Carlo Smoothing for Nonlinear Time Series. *Journal of American Statistical Association*, 99, 156-168.
- Hansen, P.R., and Lunde, A. (2006) Realized Variance and Market Microstructure Noise. *Journal of Business and Economics Statistics*, 24, 127-161.
- Harris, L. (1990) Estimation of Stock Price Variances and Serial Covariances from Discrete Observations. *Journal of Financial and Quantitative Analysis*, 25, 291-306.
- Hinamoto, T., Ohnishi, H., and Lu, W.-S. (2003) Roundoff noise minimization of state-space digital filters using separate and joint error feedback/ coordinate transformation. *IEEE Trans. Circuits Syst. I*, 50, 23-33.
- Howison, S., and Lamper, D. (2001) Trading volume in models of financial derivatives. *Applied Mathematical Finance*, 8, 119-135.
- Jacod, J., Li, Y., Mykland, P.A., Podolskij, M., and Vetter, M. (2009) Microstructure noise in the continuous case: The pre-averaging approach. *Stochastic Processes and their Applications*, 119, 2249-2276.
- Joe, H. (1995) Approximations to multivariate normal rectangle probabilities based on conditional expectations. *Journal of the American Statistical Association*, 90, 957-964.
- Kalnina, I., and Linton, O. (2008) Estimating quadratic variation consistently in the presence of endogenous and diurnal measurement error. *Journal of Econometrics*, 147, 47-59.
- Kong, A., Liu, J., and Wong, W. (1994) Sequential imputation and Bayesian missing data problems. *Journal of American Statistical Association*, 89, 278-288.
- Kotecha, J. and Djuric, P. (1999) Gibbs sampling approach for the generation of truncated multivariate Gaussian random variables. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1757-1760.
- Kristensen, D. (2009) Nonparametric Filtering of the Realised Spot Volatility: A Kernel-based Approach. *Econometric Theory*, in press.
- Large, J. (2007) Estimating Quadratic Variation When Quoted Prices Change By A Constant Increment. unpublished manuscript.
- Li, Y., and Mykland, P.A. (2007) Are volatility estimators robust with respect to modeling assumptions?. *Bernoulli*, 13, 601-622.
- Munk, A., and Schmidt-Hieber, J. (2009) Nonparametric Estimation of the Volatility Function in a High-Frequency Model corrupted by Noise. unpublished manuscript.
- Neddermeyer, J.C. (2010) Nonparametric Particle Filtering and Smoothing with Quasi-Monte Carlo Sampling. *Journal of Statistical Computation and Simulation*, in press.
- Plerou, V., Gopikrishnan, P., Gabaix, X., A Nunes Amaral, L., and Stanley, H.E. (2001) Price fluctuations, market activity and trading volume. *Quantitative Finance*, 1, 262-269.
- Podolskij, M., and Vetter, M. (2009) Estimation of volatility functionals in the simultaneous presence of microstructure noise and jumps. *Bernoulli*, 15, 634-658.

- Renfors, M. (1983) Roundoff noise in error-feedback state-space filters. *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, 619-622.
- Robert, C. (1995) Simulation of truncated normal variables. *Statistics and Computing*, 5, 121-125.
- Robert, C.Y., and Rosenbaum, M. (2008) Ultra high frequency volatility and co-volatility estimation in a microstructure model with uncertainty zones. unpublished manuscript.
- Rodriguez-Yam, G., Davis, R., and Scharf, L. (2004) Efficient gibbs sampling of truncated multivariate normal with application to constrained linear regression. Technical report, Colorado State University, 2004.
- Rosenbaum, M. (2009) Integrated volatility and round-off error. *Bernoulli*, 15, 687-720.
- Sato, M. (2000) Convergence of on-line EM algorithm. In *Proc. Int. Conf. on Neural Information Processing*, 1, 476-481.
- Titterton, D.M. (1984) Recursive Parameter Estimation Using Incomplete Data. *Journal of the Royal Statistical Society, Series B*, 46, 257-267.
- Voev, V., and Lunde, A. (2007) Integrated Covariance Estimation using High-Frequency Data in the Presence of Noise. *Journal of Financial Econometrics*, 5, 68-104.
- Wang, S., and Zhao, Y. (2006) Almost sure convergence of Titterton's recursive estimator for mixture models. *Statistics & Probability Letters*, 76, 2001-2006.
- Zeng, Y. (2003) A Partially Observed Model for Micromovement of Asset Prices with Bayes Estimation via Filtering. *Mathematical Finance*, 13, 411-444.
- Zhang, L., Mykland, P.A., and Aït-Sahalia (2005) A Tale of Two Time Scales: Determining Integrated Volatility with Noisy High-Frequency Data. *Journal of the American Statistical Association*, 100, 1394-1411.
- Zhou, B. (1996) High-Frequency Data and Volatility in Foreign-Exchange Rates. *Journal of Business & Economic Statistics*, 14, 45-52.
- Zumbach, G., Corsi, F., and Trapletti, A. (2002) Efficient estimation of volatility using high-frequency data. Technical Report, Olsen & Associates.